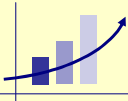


Oracle und Data Striping

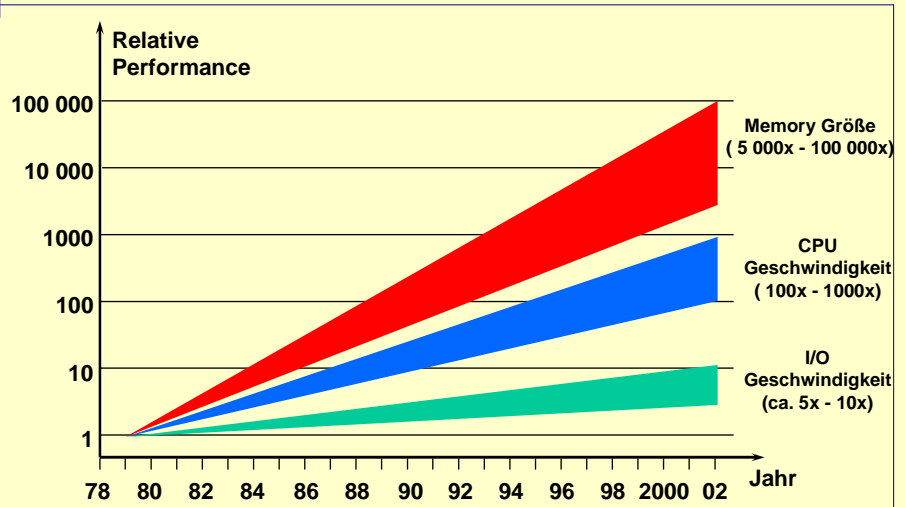
2E03

Hermann Brunner
 Angerwiese 15
 85567 Grafing
 Te | 080 92 / 328 29
 Fax 080 92 / 328 42
 hermann@brunner-consulting.de
 www.brunner-consulting.de

brunner consulting Oracle und Data Striping 1

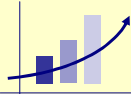


Wozu überhaupt über I/O Performance nachdenken?



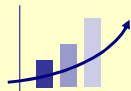
Jahr	Memory Größe (5 000x - 100 000x)	CPU Geschwindigkeit (100x - 1000x)	I/O Geschwindigkeit (ca. 5x - 10x)
78	1	1	1
80	~10	~5	~2
82	~100	~20	~3
84	~1 000	~50	~4
86	~10 000	~100	~5
88	~100 000	~200	~6
90	~1 000 000	~500	~7
92	~10 000 000	~1 000	~8
94	~100 000 000	~2 000	~9
96	~1 000 000 000	~5 000	~10
98	~10 000 000 000	~10 000	~11
2000	~100 000 000 000	~20 000	~12
02	~1 000 000 000 000	~50 000	~13

brunner consulting Oracle und Data Striping 2



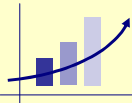
Entwicklung der I/O-Performance

	1980 (RM05)	2002	Faktor
Kapazität	300 MB	36 GB	120
Avg. Seektime	45 msec	7,5 msec	6
Drehzahl	3600 rpm	14 400 rpm	4
Avg Access Time	55 msec	10-12 msec	5
Data Rate	500 KB/sec	5-10 MB/sec	10 - 20
Max. I/O-Rate	18	80-100	5



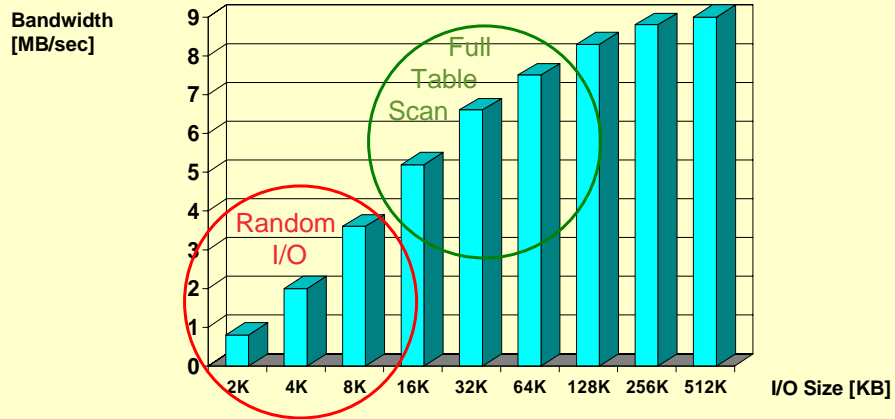
Die wichtigsten Maßzahlen

Deutsch	Englisch	Einheit
Bandbreite	Bandwidth Data Rate / Throughput	MB/sec
Datenrate Durchsatz I/O Leistung	I/O Request Rate	I/Os/sec
Mittlere Suchzeit	Average Seek Time	msec
Drehwartezeit Latenzzeit	Rotational Latency	msec
Mittlere Zugriffszeit Mittlere Antwortzeit	Average Access Time Average Response Time	msec



Bandbreite versus I/O-Größe

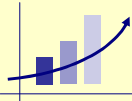
9,1 GB / 10.000 rpm IBM-Disk am Adaptec 2940 VW-Controller
Sequential Reads, EZ-SCSI Benchmark



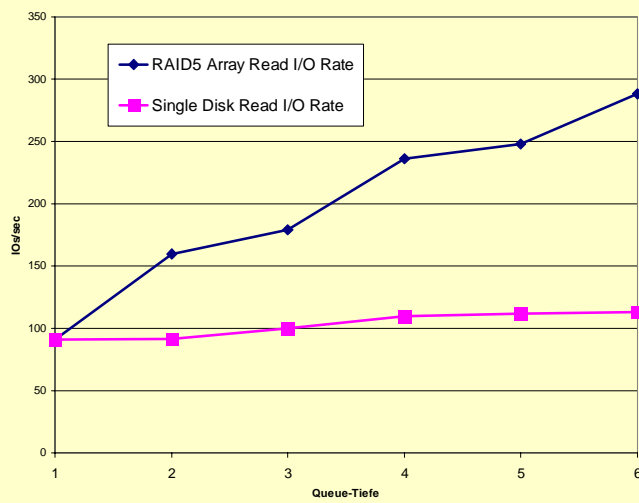
brunner consulting

Oracle und Data Striping

5



I/O-Durchsatz

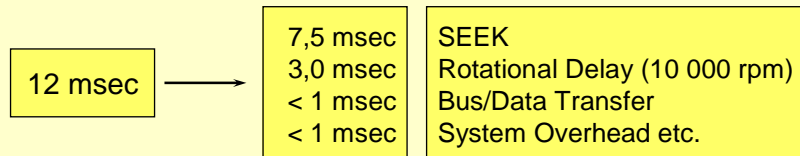


brunner consulting

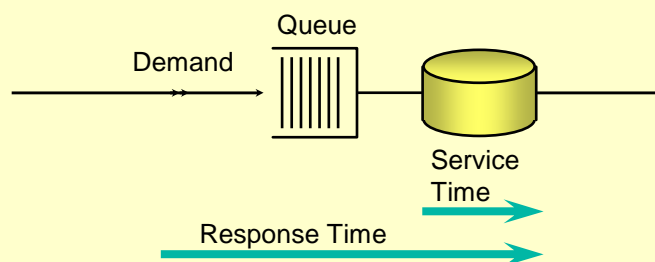
Oracle und Data Striping

6

Zusammensetzung der Mittleren Zugriffszeit



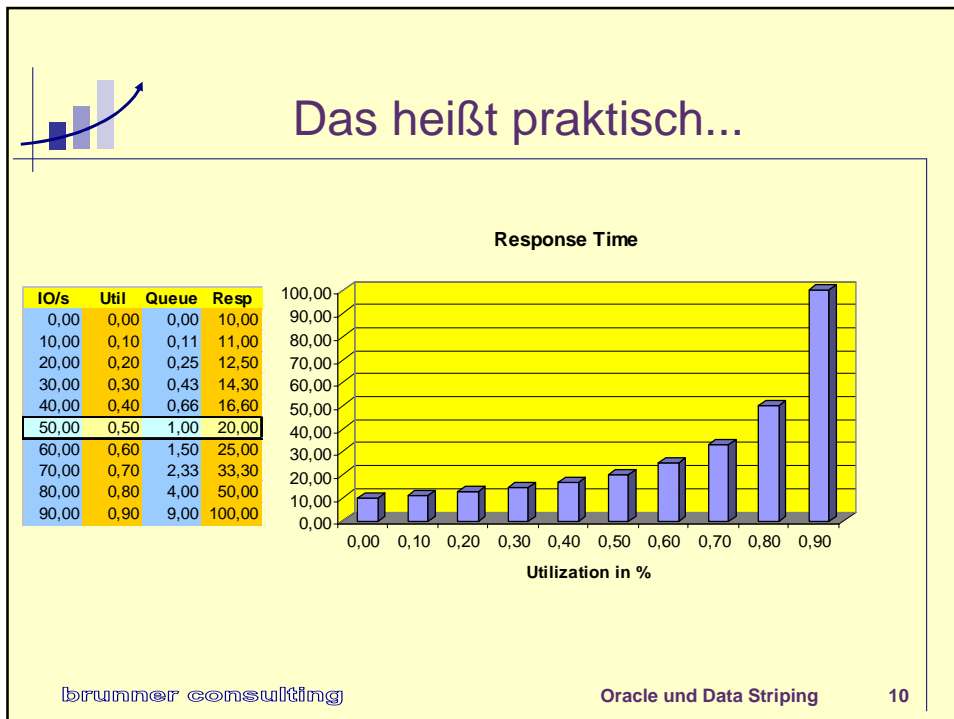
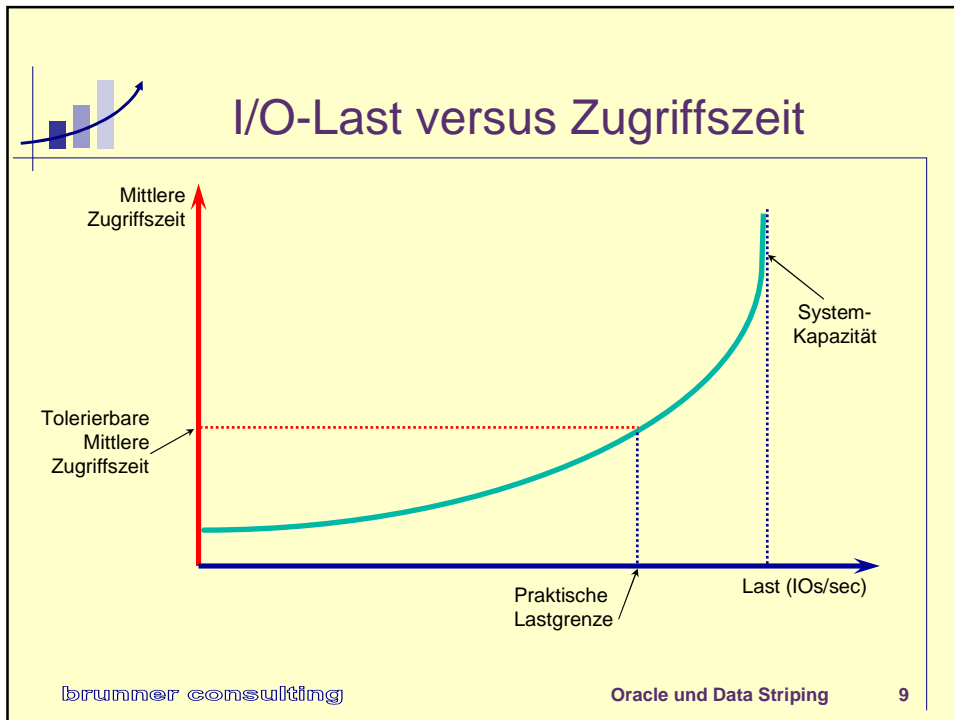
Queueing Theorie

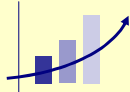


$$\text{Utilization} = \text{Service Time} * \text{Demand}$$

$$\text{Queue} = \text{Utilization} / (1 - \text{Utilization})$$

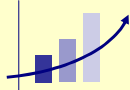
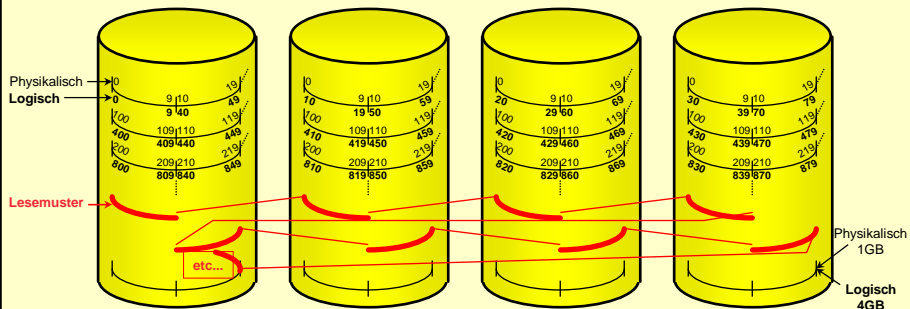
$$\text{Response Time} = \text{Service Time} * (1 + \text{Queue})$$





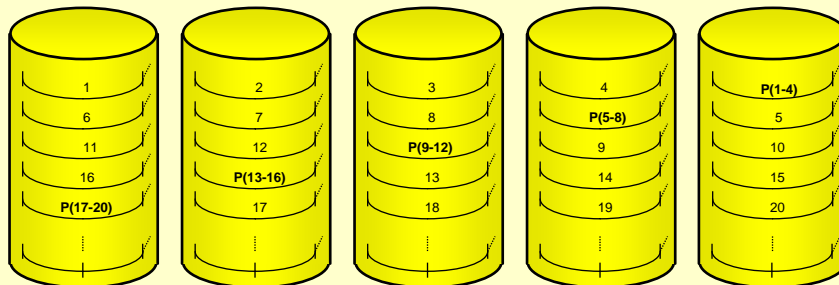
RAID 0

Beispiel: Chunk Size = 10 Sectors
Track Size = 100 Sectors

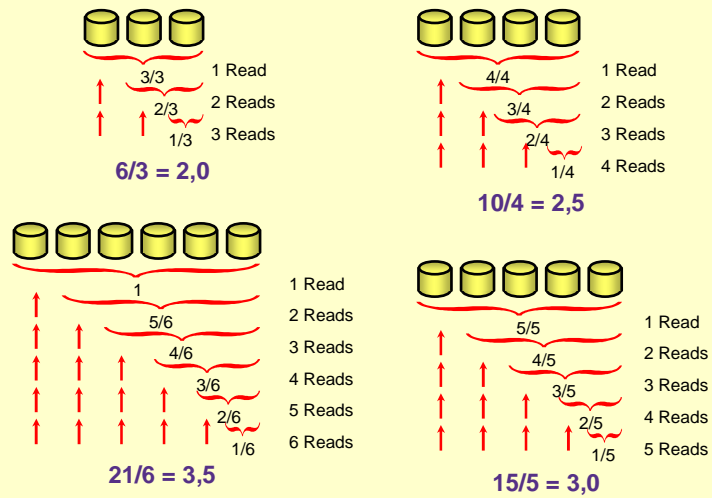


RAID 5

Chunk Size = 1 Track oder größer
Rotierende Parity-Information



Wahrscheinlichkeit gleichzeitiger Read-Operationen



brunner consulting

Oracle und Data Striping

13

Performance der RAID Levels

	Transaktions-I/O		Große File I/O	
	Read	Write	Read	Write
JBOD	OK	OK	OK	OK
RAID 0 (Strip)	Sehr gut	Sehr gut	Gut	Gut
RAID 1 (Shad)	Gut	OK	OK	OK
RAID 0+1	Exzellent	Sehr gut	Gut	OK
RAID 3	Schlecht	Schlecht	Sehr gut	Sehr gut
RAID 5	Sehr gut	Schlecht	OK	OK

brunner consulting

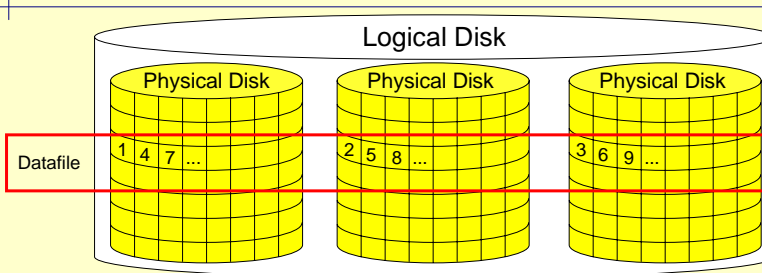
Oracle und Data Striping

14

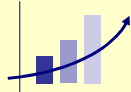
Einfluß von Read und Write Caches

	XACTION I/O (Small)		STREAM I/O	
	Read	Write	Read	Write
JBOD	OK	Sehr gut	OK	Gut
RAID 0 (Strip)	Sehr gut	Exzellent	Gut	Sehr gut
RAID 1 (Shad)	Gut	Sehr gut	OK	Gut
RAID 0+1	Exzellent	Exzellent	Gut	Sehr gut
RAID 3	Schlecht	Gut	Sehr gut	Exzellent
RAID 5	Sehr gut	Sehr gut	OK	Gut

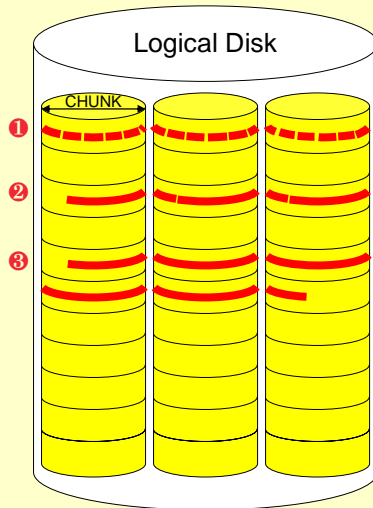
Disk Striping



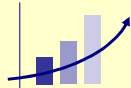
- Datafile ist aus Oracle-Sicht nicht gestriped ...
- Auf physikalischer Ebene Striping auf CHUNK Grenzen
 - ⇒ Also unabhängig von Oracle-TS-Parametrisierung! (Extents, DB-Block-Grenzen)



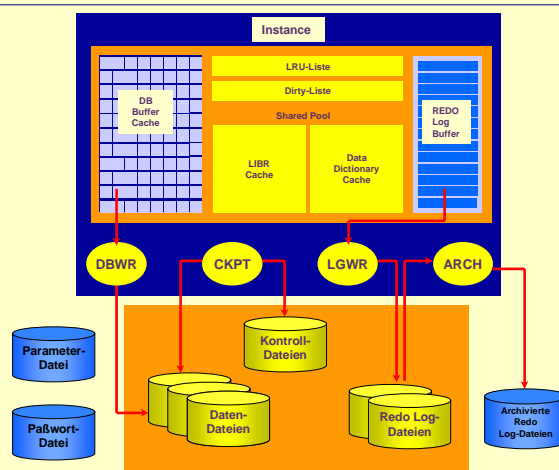
DB-Block Size ↔ CHUNK Size

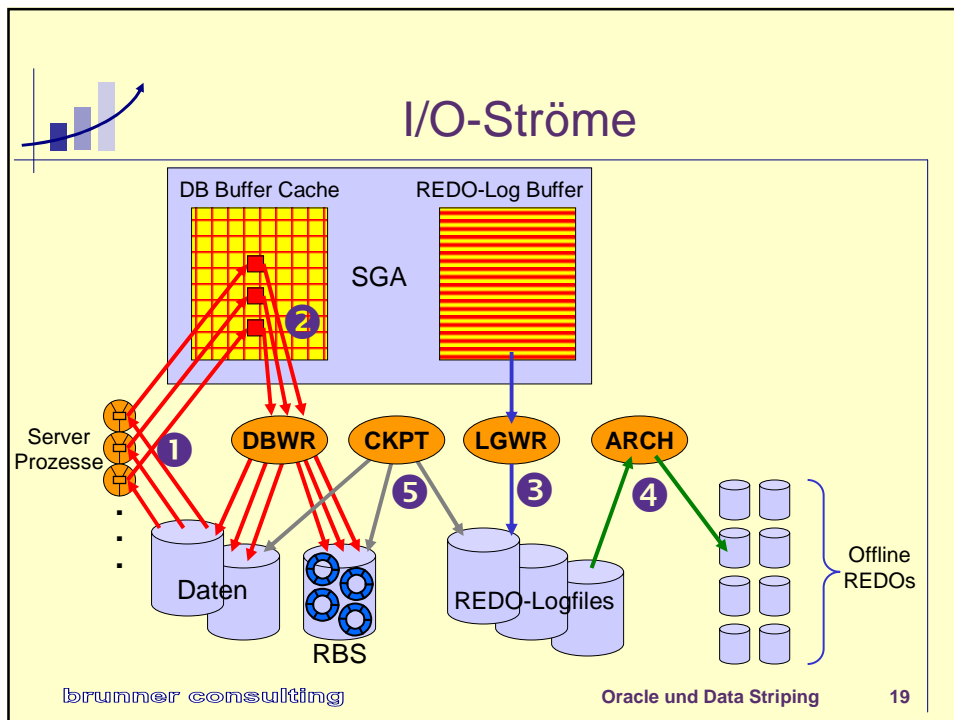


- 1** CHUNK deutlich größer als DB-Block
(z.B. CHUNK=200 Blocks, DB-Blocks 2K-8K)
⇒ Bei den meisten DB-Blocks muß zum Lesen nur eine Disk "angefaßt" werden
- 2** CHUNK etwa gleich groß wie DB-Block
⇒ Jeder DB-Block geht über 2 Disks (fehlendes Alignment!)
- 3** DB-Blocks deutlich größer als CHUNK
⇒ Jeder DB-Block liegt auf mehreren Disks

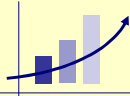


Oracle-Architektur





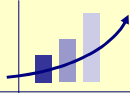
- ## Es entstehen folgende I/O-Ströme
- 1** Viele parallele Server-Prozesse
 Lesen aus den Datafiles
 (Zeitkritisch → geringere Response Time gefragt)
 - 2** DBWR - schreibt nahezu kontinuierlich in Datafiles
 (Daten + RBS-Segmente → nicht zeitkritisch)
 - 3** LGWR - schreibt spätestens bei jedem Commit in die REDO-Logfile → zeitkritisch
 - 4** ARCH - liest REDO-Log, schreibt archivierte Offline-REDOs → nicht zeitkritisch, aber minimaler Durchsatz muß erfüllt sein
 - 5** CKPT - schreibt gelegentlich zu CTL-Files (→ nicht zeitkritisch)
- brunner consulting Oracle und Data Striping 20



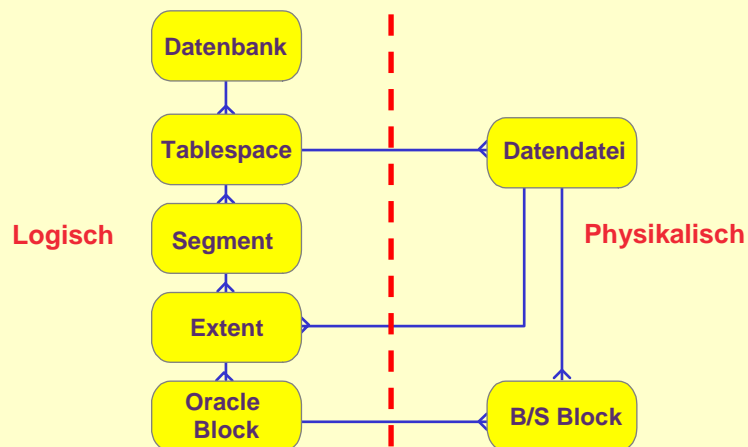
Konfigurationsregeln

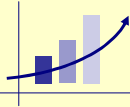
② Data Files

- ⇒ I/O-Aufkommen hängt von Verhalten der Anwendungen ab
- ⇒ Last-Verteilung ist wichtig
 - * Häufig werden nicht ausreichend viele Data Files konfiguriert
- ⇒ Faustregel:
 - * Keine Data File sollte > 50% der maximalen I/O-Last der zugrundeliegenden Hardware "abbekommen"
- ⇒ Lösungen: "Fleißiger DBA" oder Striping / RAID 5



Datenbank-Strukturen

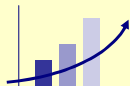




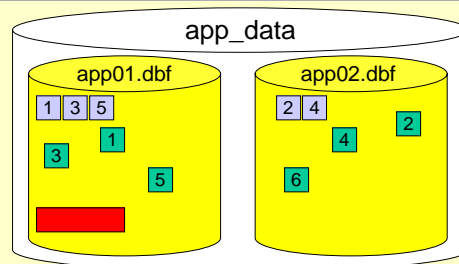
Tablespaces anlegen



Beispiel

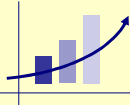
```
CREATE TABLESPACE app_data
DATAFILE '/DISK4/app01.dbf' SIZE 100M,
         '/DISK5/app02.dbf' SIZE 100M
MINIMUM EXTENT 500K
DEFAULT STORAGE (INITIAL 500K NEXT 500K
MAXEXTENTS 500 PCTINCREASE 0);
```



Verteilung der Extents



- Extents werden ähnlich einem "Striping-Muster" auf Data Files verteilt.
 - ⇒ Vorteil: Lastbalancierung → 
 - ⇒ Gefahr: Fragmentierung → 



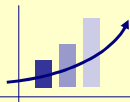
Verteilung der Extents

Aber:

```
CREATE TABLE XYZ (coll typ1,...)
TABLESPACE app_data
STORAGE (initial 10M next 10M) → ■
```

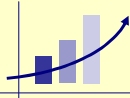
Warum? MINEXTENTS = 1 (=default)

VORSICHT bei EXP/IMP mit "COMPRESS" Option



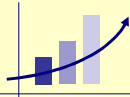
Generell

- **Beide** Methoden
(RAID, Oracle-TS-Striping)
bringen bei richtiger Konfiguration und
Nutzung eine Lastverteilung
 - ⇒ I/O **Leistung** steigt
 - ⇒ Response Time einzelner I/Os bleibt gleich
 - ⇒ Es sei denn, wir haben Queuing



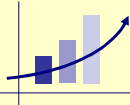
Vorteile / Nachteile

- Beide Striping Methoden
 - ⇒ bringen Vorteile bei hoher Durchsatz-Anforderung, die Einzeldisk überfordern würde
 - ⇒ können nur positive Wirkung entfalten, wenn auch wirklich parallele IOs zur Ausführung kommen! (Queue >> 1.0)
 - ⇒ bringen keine Vorteile bei sequentiellm IO (→ FULL TABLE SCAN!)



Unterschiede

- RAID Controller sind "transparent", d.h. auf Operating System / File System / DB-Ebene ist nichts vom Striping "zu sehen" → Kein "Management" nötig
- Tablespace Striping unterliegt der vollen Kontrolle des DBA bzw. Entwicklers.
 - ↑ Dynamische Erweiterungen möglich
 - ↑ Effekte auf TS- oder Objekt-Ebene steuerbar
 - ↓ Management-Aufwand
 - ↓ Monitoring nötig
- "Arbeit" liegt in "Hardware" bzw. "Software"
Somit → theoretisch Auswirkungen auf Systemlast
Jedoch → praktisch kaum nachweisbar



Was sollte man wählen?

● Fragen

- ⇒ Wo liegt unser Know-How?
 - * Mehr System / HW Know-How → RAID
 - * Mehr DBA / DB Know How → TS-Striping
- ⇒ Wie gut "durchblicken" wir
 - * Die Anwendung / Datenbank
 - ✓ Analyse **unbedingt** notwendig!!!

● Faustregeln

- ⇒ HW-Striping ist einfacher, HW ist teurer, man kann weniger falsch machen ("set and forget")
- ⇒ TBSpace-Striping muß professionell "gemanaged" werden (→ Manpower / Zeitaufwand !)