

第10章 数据库恢复技术

备份与恢复技术？

文件管理系统

以文件为单位，涉及版本、快照、副本。。。

方法：Ghost、恢复出厂设置、副本文件、操作系统补丁更新时的备份、备份软件（例如：云备份）

数据库管理系统

细化到数据结构（项）、备份涉及数据变化的过程，数据的取值对应任务的状态

方法：备份+日志、镜像、备份软件

应用背景：

DBMS——提供数据共享和管理服务的系统软件，数据动态更新、大量并发访问的用户

故障的特征，数据的问题（内存/外存、干净/脏、。。。)

DBMS的备份与恢复

- 正确性的标准？（不“丢失”数据？）
- 面临的问题
 - 1) 正确；
 - 2) 智能（自我保护的程序机制）；
 - 3) 故障的种类多（应用程序出错、网络断了、操作系统蓝屏、宕机、硬盘坏了。。。），响应机制不同；
 - 4) 性能（备份的开销、恢复的开销、服务的性能）。
- 学习内容
 - 备份与恢复的基本原理、系统内部的协议、优化机制
- 问题复杂的原因
 - 1) **DBMS**内部程序结构及其执行的复杂性（多任务）；
 - 2) 应用的业务逻辑需求。

事务

10.1 事务的基本概念 (transaction)

1、定义



构成一个独立逻辑工作单位的数据库操作集。

事务——transaction——交易、买卖

现代计算机的运行原理及硬件特点使得事务处理任务艰巨，DBMS提供数据共享服务的需求使得事务处理更加复杂。



事务处理成为数据库的核心问题之一，标志性人物：Jim Gray。

事务的定义形式



➤ 一条SQL语句;

例： 将2号课程的成绩记录增加10分

➤ 一组SQL语句序列

例： （1） 将2号课程的成绩记录增加10分;

（2） 在应用日志表中增加描述上述操作的一条记录;

➤ 一个包含对数据库操作的应用程序

例： 可能多条SQL语句，存在于不同的程序分支。

2、事务的构成方式

① 显式

BEGIN TRANSACTION

• • •

COMMIT 或者 ROLLBACK



其中：

COMMIT：提交，事务对DB修改写回到磁盘上的DB中去。

ROLLBACK：回滚，撤消对DB之修改，回滚到事务开始状态。

ABORT???——底层实现技术

② 隐式（可能是系统默认方式）

某种环境或者程序中的一条SQL语句、
应用程序或操作窗口退出

3、事务的ACID性质

1) 原子性 (Atomicity)

① 定义

事务是一个不可分割的工作单元，其对**DB**的操作要么都做，要么都不做。

② 目标

保证**DB**数据的正确性（例如：所有员工涨工资、转帐、售票的事务不能只做一部分动作）。

③ 技术

日志+**ROLLBACK (UNDO)**（意外终止）、影子数据；
并发控制（隔离保护）。

原子性需要依靠**DBMS**内部的自动保障机制。

2) 一致性 (Consistency)

① 定义

事务的执行必须是将数据库从一个正确（一致）状态转换到另一个正确（一致）状态。

例1：一个人的工龄不能大于年龄，工龄的增长和年龄的增长必须一致的、配套的修改。

例2：所有员工工资的公积金应该依据政策同步调整。

例3：转帐问题，A有100万人民币是一个正确状态，减去50万，B帐上相应增加50万，数据库从一个正确状态转变另一个正确状态。

这两个操作，若只做其中一个，则不能实现数据库从一个正确状态转到另一个正确状态，破坏了事务一致性。

例4：将数据集合划分为三个子集，分三次读取三个子集的数据并进行小计和总计（一个事务内部的多次相关的读写操作，内容之间在全局逻辑上应该是相容的、一致的）。

2) 一致性 (consistency)

② 目标

保证DB数据正确性（防止丢失更新、读脏、读不可重复）。

③ 技术

并发控制、恢复机制

④ 实现

用户定义事务（保证相关操作在一个事务中）；

DBMS负责维护事务执行导致数据库状态变化过程中的一致性。

3) 隔离性 (isolation)

① 定义

一个事务中对数据库的操作及使用的数据与其它并发事务无关，并发执行的事务间不能互相干扰。

② 目标

避免链式干扰。

③ 技术

并发控制。

④ 实现

DBMS依据应用程序设定的事务隔离级别自动实现。

插入

更新

更新

4) 持久性 (Durability)

① 定义

一个已提交事务对数据库的更新是永久性的，不受后来故障的影响。


② 目标

保证数据库可靠性

③ 技术

提交持久（内存是挥发装置，外存是抗挥发装置）。

（事务终止前应完成commit）



外存!

备份 + 日志。

④ 实现

持久性需要依靠DBMS的恢复子系统。

ACID特性带来的DBMS技术需求

恢复:

复杂系统如何保存数据（记录过程）、恢复策略（算法）、高可用性、其他技术手段

并发:

锁、协议、标准

10.2 数据库恢复概述

将因破坏或故障而导致的数据库数据的错误状态恢复到最近一个正确状态的技术。

目标

- 1、保持事务原子性（Atomicity）；
- 2、保持事务持久性（Durability）。



背景：事务——transaction——交易、买卖

10.3 数据库系统故障

1、事务故障

1) 表现形式

①应用处理异常

可能产生自程序**预留的**异常情况的应对方案。

更多的故障来自于**非预期的**，是不能由应用程序处理的。✦

*断网、应用程序进程僵死、应用程序进程被意外杀死、
应用程序端电脑死机、断电*

②系统异常

事务超时、死锁、活锁等



2) 事务故障的特征

- ① 特定的事务没有到达预期的终点（COMMIT），事务夭折；
- ② 夭折事务对数据库的部分修改可能已写入数据文件。

（数据库可能因此处于不正确、不一致状态）

例：理财产品交易中，客户的理财账户余额已减少，但理财产品的购买记录还未来得及保存到数据文件，此时事务发生了异常。

事务的ACID特性，原子性A，一致性C。

2、系统故障

1) 表现形式

① 特定类型的硬件故障（CPU、内存、主板等**非外存储**设备）；

② 系统软件故障

DBMS: ORACLE、SQL SERVER、MYSQL、DB2、。。。

OS: UNIX、WINDOWS、LINUX、。。。

死机

蓝屏

意外重启

某系统功能意外退出

。。。

③ 系统操作失误：非正常关机/重启、强行终止系统进程、意外卸载相关系统运行环境。。。

④ 系统异常断电（**重启之后系统未发现数据库的存储文件错误或者磁盘错误**）

2) 特征

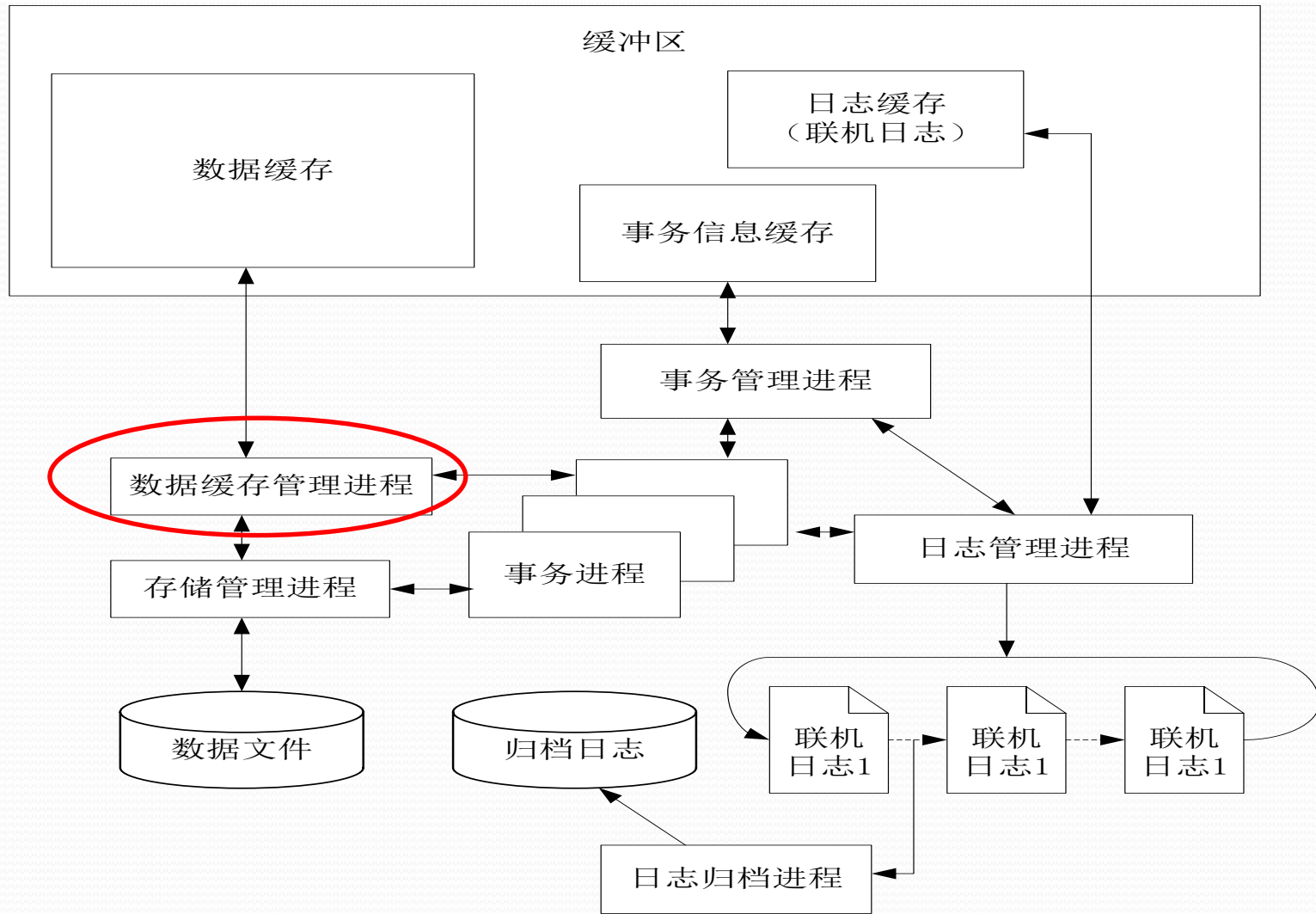
- ① 内存数据丢失或不再可靠;
- ② 外存数据未受到破坏;
- ③ 一些尚未完成事务的更新结果可能已写入数据库存储介质;
- ④ 已完成事务的更新结果可能部分还未写入数据库存储介质（数据文件，也可能正处于提交过程之中）;
- ⑤ 已完成事务的结果可能全部未写入数据库（例如正在等待检查点）。



数据库的数据存储介质依然可靠，但是数据处于不正确或不一致状态（包括破坏了持久性）



DBMS的缓存与核心进程



3、介质故障

1) 分类

① 磁盘故障

磁盘损坏（磁道、扇区、分区、文件分配信息。。。），
磁盘读写装置损坏（磁头、电机。。。）。

o 数据库属性 - course

选择页

- 常规
- 文件
- 文件组
- 选项
- 更改跟踪
- 权限
- 扩展属性
- 镜像
- 事务日志传送

脚本 帮助

数据库名称 (N): course

所有者 (O): 2011-20121217JJ\Administrator

☒ 使用全文索引 (U)

数据库文件 (F):

逻辑名称	文件类型	文件组	初始大小(MB)	自动增长	路径
course	行数据	PRIMARY	3	增量为 1 MB, 不限制增长	...
course_log	日志	不适用	1	增量为 10%, 增长的最...	...
course2	行数据	PRIMARY	3	增量为 1 MB, 不限制增长	...

② 外界干扰

强磁场干扰（磁性数据被清洗），灾害。

2) 特征

外存数据库中的数据部分或全部丢失，数据存储文件本身被破坏。

目前，成熟的DBMS软件一般能够在服务启动时校验存储介质上的数据文件是否异常。

Tablespace Details

<input type="checkbox"/>	Tablespace Name	Total allocated bytes	Total Allocated Bytes (MB)	Used Bytes (MB)	Free Bytes (MB)	Percentage of Used Bytes (%)	Percentage of Free Bytes (%)	Allocated Blocks	Free Blocks	Health	Configure Alarms
<input type="checkbox"/>	TEMP	<div><div></div></div>	235	0	235	0	100	30,080	0	<div><div></div></div>	<div><div></div></div>
<input type="checkbox"/>	USERS	<div><div></div></div>	11,264	2.56	11,261.44	0.02	99.98	12,800	12,472	<div><div></div></div>	<div><div></div></div>
<input type="checkbox"/>	UNDOTBS1	<div><div></div></div>	32,767.98	82	32,685.98	0.25	99.75	10,880	384	<div><div></div></div>	<div><div></div></div>
<input type="checkbox"/>	SYSAUX	<div><div></div></div>	32,767.98	490.13	32,277.86	1.5	98.5	69,120	6,384	<div><div></div></div>	<div><div></div></div>
<input type="checkbox"/>	TESTSPACE	<div><div></div></div>	10	1	9	10	90	256	128	<div><div></div></div>	<div><div></div></div>
<input type="checkbox"/>	SYSTEM	<div><div></div></div>	2,000	1,789.25	210.75	89.46	10.54	256,000	26,976	<div><div></div></div>	<div><div></div></div>

Action Compare Reports

Tablespace Status

TI

TI

U

U

S

S

TI

Action

--Se

Performance of Data Files

<div></div>	Data File Name	Tablespace Name	Status	DataFiles AutoExtend	Created Bytes (MB)	Reads	Writes	Average Read Time (ms)	Average Write Time (ms)	Health	Configure Alarms
<div></div>	C:\ORACLE\EXE\APP\ORACLE\ORADATA\SYSTEM.DBF	SYSTEM	SYSTEM	YES	0	4,137	85	1	1	<div></div>	<div></div>
<div></div>	C:\ORACLE\EXE\APP\ORACLE\ORADATA\TESTSPACE4.DBF	TESTSPACE	ONLINE	YES	2	6	2	1	0	<div></div>	<div></div>
<div></div>	C:\ORACLE\EXE\APP\ORACLE\ORADATA\USERS.DBF	USERS	ONLINE	YES	0	6	2	1	0	<div></div>	<div></div>
<div></div>	C:\ORACLE\EXE\APP\ORACLE\ORADATA\SYSAUX.DBF	UNDOTBS1	ONLINE	YES	0	27	548	10	1	<div></div>	<div></div>
<div></div>	C:\ORACLE\EXE\APP\ORACLE\ORADATA\UNDOTBS1.DBF	SYSAUX	ONLINE	YES	0	984	11	1	2	<div></div>	<div></div>

Compare Reports

--Select Metric--

4 计算机病毒

1) 表现形式

① 消耗资源

内存、磁盘、网络端口，破坏系统的正常运行

② 泄露信息

系统信息、数据库信息

③ 篡改数据

④ 篡改程序

植入木马，埋下隐患

2) 特征

计算机病毒是一种人为造成的技术故障或破坏，含有非法或者恶意企图，在执行某个功能时启动病毒代码，可能造成对数据库系统的危害。

系统不再可靠、可信。



数据块：数据库系统中，数据库的存储单位一般是**数据块（物理块）**，数据块可能包含多个数据记录、数据项。

缓冲块：当数据库规模较大时，不可能所有的数据库内容驻留内存，一般选取事务操作需要访问的数据块驻留内存，主存中的数据块称为**缓冲块（缓存页面）**。

磁盘缓冲区：DBMS中有集中存放缓冲块的内存区域，在本章后续讨论中称为磁盘缓冲区，简称缓冲区、缓存。

因为数据访问、缓存有限等原因，磁盘和缓存间有**块移动**，对应的两个操作：

input (B)：读取物理块**B**到缓存。

output (B)：将缓存中的缓冲块**B**覆盖写到磁盘上的物理块。



事务和数据库的交互：交互是通过read或write操作完成的。在执行这两个操作时，若数据X所在块 B_x 不在缓存中，会触发执行input（ B_x ）。

缓冲块写到磁盘（output操作）的原因：

- 缓冲区管理器基于缓存使用策略调度页面（内存资源有限）、
- 数据库系统强制输出（force-output（），例如检查点、转储、关机、。。。）

Write（X）和output（ B_x ）操作之间的关系：

二者不必紧密相邻，可能write操作后过一段时间才真正执行output操作。

存在的问题：在write（X）和output（ B_x ）操作之间系统可能崩溃，此时的故障恢复需要分析针对X该采取何种动作。



各类故障对数据库的影响

- 故障发生时数据可能不正确（事务的运行被恶意干扰或非正常终止）。

当涉及多个**output**操作的事务出现故障时，如果只知道数据本身的当前值状态**而无相关事务信息**，是无法判断哪些值是完成了相应的**output**操作的。需要借助系统的容错机制，**找出不正确的数据，恢复正确的数据**。

- 不同的故障影响的范围不同，采取的恢复策略也不尽相同。
自动、人工启动、借助外部资源

- 数据文件本身可能被破坏
需要去寻找可用的数据，重建系统状态。

- 恢复的基本原理：冗余（包括对于数据变化过程的记录）

10.4 恢复技术

备份 + 日志

10.4.1 备份技术

1、备份方式

1) 静态备份

——数据库系统中无事务运行时进行转储（dump）。

① 特征：

- 转储期间不对数据库进行任何操作；
- 得到一个一致性付本。

② 优点——简单

③ 缺点：停止一切事务运行；降低数据库可用性。



2) 动态备份

转储与事务并发执行。

① 特征

转储期间可对数据库进行存取与修改操作。

② 优点

不影响事务运行。

时刻	事务1	事务2
转储A时	A=100,B=200	
转储B时		A=200,B=100

③ 缺点

获得一致性副本较麻烦。

如转储A时，其值为100，B值为200，但在随后转储B时，但另一事务修改数据库为A=200，B=100，这样备份副本是与DB中实际值不一致的过时数据。

2、备份策略

1) 海量备份

① 方法：定期或不定期将数据库全部数据转储。

② 优点：简单。

③ 缺点：重复转储；

转储量大；

停止运行（多为静态转储）。



2) 增量备份 (incremental clumping)

① 方法：每次转储上次转储后更新过的数据。

② 优点：备份量小。

③ 缺点：恢复过程较复杂。

(完全、

累积——自上次完全备份或者累积备份之后的修改、

增量——自上次完全或者累积或者增量备份之后的修改)



3) 写副本

① 方法：每次写时，同时写另一个副本。

② 优点：简单。

③ 缺点：重复写，操作效率下降。

10.4.2 日志 (logging)

HUST-CS

PANPENG

某工厂机组值班日志

日志本页号

列表 日志 参数记录 事件 岗位分派 指令 预览

事件

值班日志本: #4机值班员日志

日志本页号: 4519 2009

状态: 当前

倒班开始: 2009-07-08 14:00:00

倒班结束: 2009-07-08 20:00:00

班组人员:

任务区域: ---请选择---

事件类别: 运行操作

受影响的范围

设备编码: 消缺单: 措施:

位置: 内容:

事件

开始时间: 2009-06-16 14:38:50

结束时间:

文档

事件类别

事件描述

事件类别

受影响设备

受影响位置

值班日志本	事件开始	事件类别	受影响设备	受影响位置
#4机值班员日志	2009-06-16 14:38	运行操作		
#4机值班员日志	2009-06-16 14:53:00	运行操作		
#4机值班员日志	2009-06-16 19:41:29	缺陷登记		
#4机值班员日志	2009-06-16 19:42:24			
#4机值班员日志	2009-06-16 19:44:12	缺陷记录		
#4机值班员日志	2009-06-16 19:45:14	工作票		
#4机值班员日志	2009-06-16 19:45:41	工作票		

增加 删除 修改 复制 确定 取消 打印

某电站通讯设备监控日志

田湾河流域梯级电站通讯设备监控系统

历史告警 用户管理 值班日志 综合报表 用户登陆 用户注销 退出监控

田湾河监控系统
大发站
通信电源
电池巡检仪
中心通信
交换机1
交换机2
+ 金窝站
+ 仁宗海站
+ 营地站
+ 引田入环首部
+ 仁宗海首部

记录时间	记录类型	操作员名称	名称	新值	旧值
2010-8-16 17:01:06	登录成功	SysAdmin		---	---
2010-8-16 17:02:00	操作	SysAdmin	报警声音_特急消音	True	False
2010-8-16 17:02:00	操作	SysAdmin	报警声音_一般消音	True	False
2010-8-16 17:02:01	操作	SysAdmin	报警声音_不急消音	True	False
2010-8-16 17:02:01	操作	SysAdmin	报警声音_紧急消音	True	False
2010-8-16 17:02:02	操作	SysAdmin	报警声音_一般消音	False	True
2010-8-16 17:02:02	操作	SysAdmin	报警声音_特急消音	False	True
2010-8-16 17:02:02	操作	SysAdmin	报警声音_紧急消音	False	True
2010-8-16 17:02:03	操作	SysAdmin	报警声音_不急消音	---	---
2010-8-16 17:02:03	操作	SysAdmin	报警声音_一般消音	---	---
2010-8-16 17:02:03	操作	SysAdmin	报警声音_紧急消音	---	---
2010-8-16 17:02:04	操作	SysAdmin	报警声音_不急消音	---	---
2010-8-16 17:02:04	操作	SysAdmin	报警声音_特急消音	True	False
2010-8-16 17:02:05	操作	SysAdmin	报警声音_紧急消音	False	True
2010-8-16 17:02:05	操作	SysAdmin	报警声音_不急消音	False	True
2010-8-16 17:02:06	操作	SysAdmin	报警声音_一般消音	False	True
2010-8-16 17:02:06	操作	SysAdmin	报警声音_特急消音	False	True

事件记录的数量: 17 新记录出现的位置: 后面

报警时间 恢复时间 应答时间 报警源 名称 记录类型 报警类型 报警区

报警的数量: 0 新报警出现的位置: 前面 允许应答

2010-08-16 17:02:21 SysAdmin

特急 紧急 一般 不急

新值与旧值

某市场经理值班日志

事件处理状态：
正在处理、
处理完毕

市场经理值班日志

值班日志 编辑 市场值班记录 月份: 2006 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2

10.4.2 日志 (logging)

实际生活中的日志——对某个设备、某种资源的动态变化过程的历史记录，以便对曾经发生的问题进行分析

——故障分析

——安全性分析

。 。 。

时间、先后顺序

地点

是否正常完成

旧状态和新状态

在数据库系统中，数据库文件也是一种设备和资源，也有类似的需求，对应的有日志文件及其维护机制。

有了日志，数据库的恢复子系统就有了“自己的数据”。

10.4.2 日志 (logging)

1、日志概念

——记录事务对数据库更新操作的文件称之为日志文件。



2、日志文件类型

- 1) 以记录为单位的日志文件;
- 2) 以数据块为单位的日志文件。

两种日志都对
应数据库操作

3、以记录为单位的日志文件内容

- 1) 事务开始标记 (一个日志记录);
- 2) 事务结束标记 (一个日志记录, 提交/撤销);
- 3) 每个事务的所有更新操作 (每个操作一个日志记录)。

如何描述“操作”? 事务夭折有无“操作?”

每个**日志记录内容**:

- 1) 事务标识 (TRID) ;
- 2) 操作类型 (插入/删除/修改) ;
- 3) 操作对象标识;
- 4) 更新前数据旧值;
- 5) 更新后数据新值。

记录头+记录体

4、以数据块为单位的日志文件内容

事务标识+数据块

- 1) 数据块 (整块) 更新前内容;
- 2) 数据块更新后内容。



5、影子拷贝与影子页面

影子拷贝是一种数据库的更新方式，要更新数据库的事务先创建数据库的一个完整拷贝，所有更新在该拷贝上进行。

数据库指针：影子拷贝中使用一个指针来标识数据库的当前拷贝，称为数据库指针，该指针存放于磁盘上。

影子拷贝事务的执行

- 若事务半途中止，仅需删除新生成的拷贝，旧版数据库拷贝不受影响。
- 若事务提交：
 - （1）先将新拷贝的所有页面写到磁盘；
 - （2）完成上述这一批写操作后，更新数据库指针使其指向新版的拷贝（更新后的**数据库指针写到磁盘上**时意味着事务提交完成）；
 - （3）更新完成后，可删除旧拷贝。



影子拷贝事务的原子性保障机制：依赖于对数据库指针的写操作的原子性，该原子性来源于磁盘系统对单个块（单个磁盘扇区）的原子性更新机制。

只要能保证数据库指针处于单个磁盘块（或者单个扇区），即可实现影子拷贝事务的原子性。

影子拷贝模式可用于小型的数据库，也普遍用于正文编辑器，但对于大型数据库则开销过大，此时可采用影子拷贝方法的一个变种——影子页面，以减少拷贝的开销。



影子页面

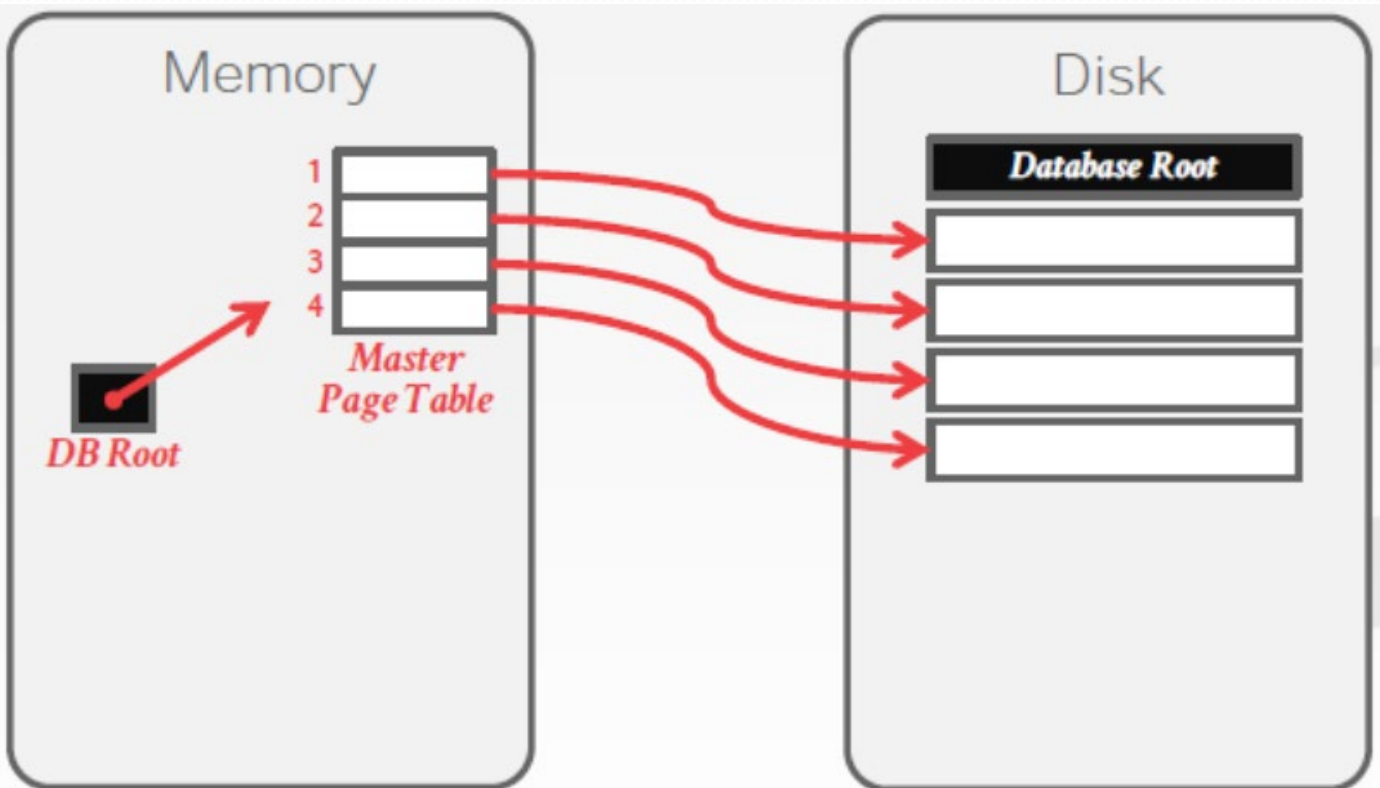
相关概念：页表，一个包含指向所有页面的指针的列表，作用和数据库指针相同。

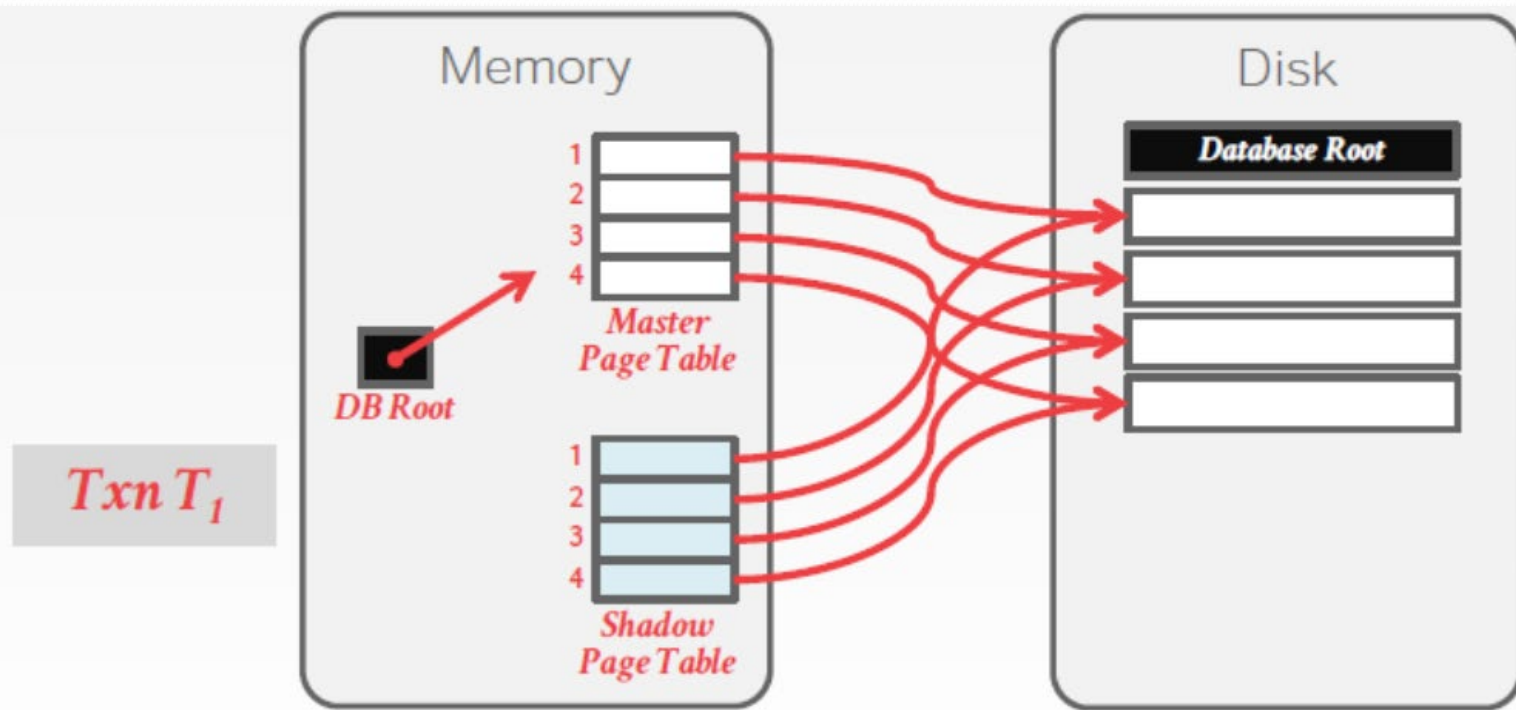
影子页面机制在页面拷贝时，只将**页表**和**所有更新的页面**（**只对增量生成影子**）拷贝到一个新位置，当提交事务时，原子性的更新指向页表的指针以指向新拷贝。

影子页面的局限性：对并发事务支持较弱，在数据库中未广泛使用。



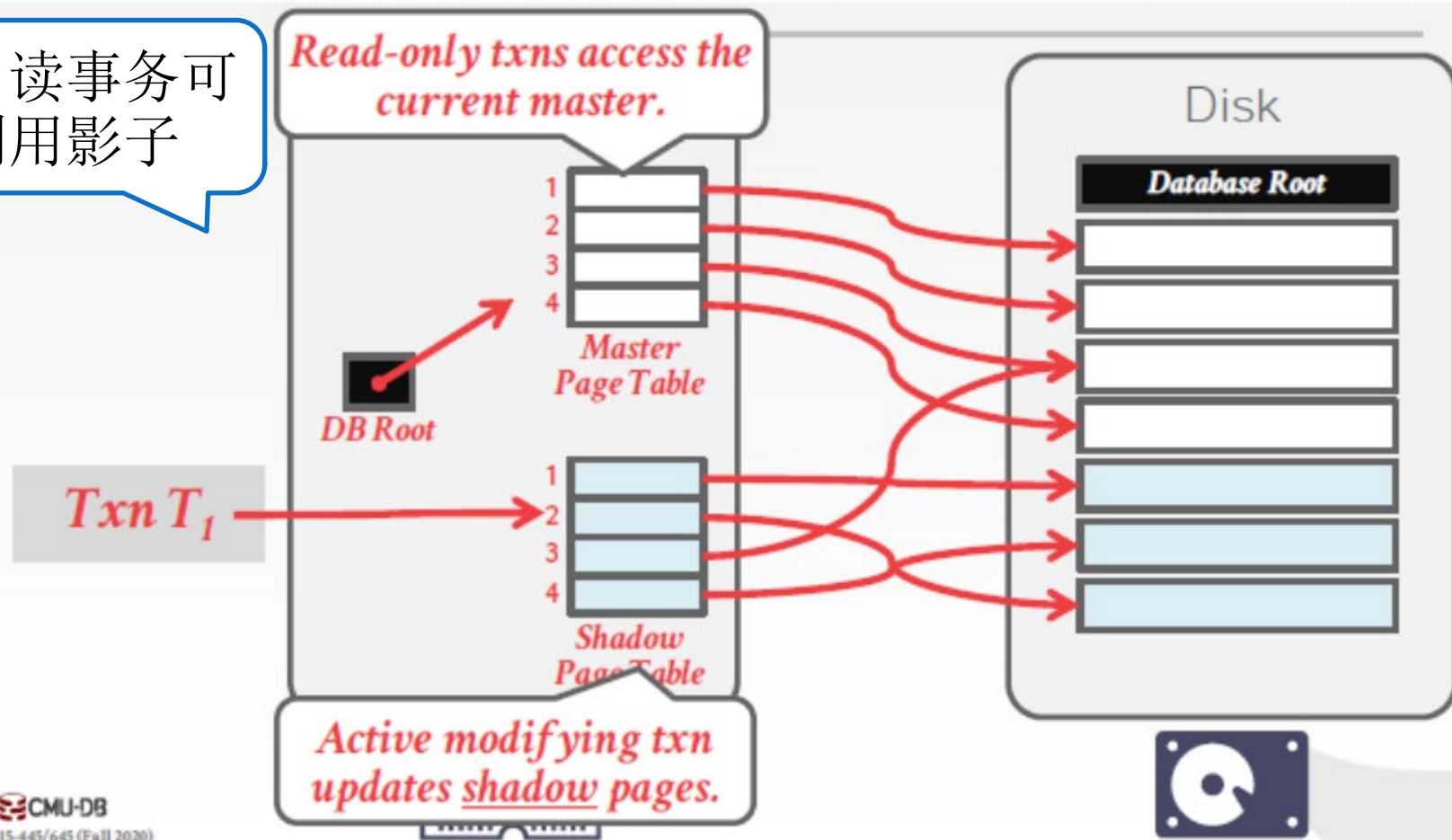
例：影子页面







只读事务可利用影子





Read-only txns access the current master.

更新事务
提交后旧
影子失效

Txn T_1
COMMIT

DB Root

1
2
3
4

Master
Page Table

1
2
3
4

Shadow
Page Table

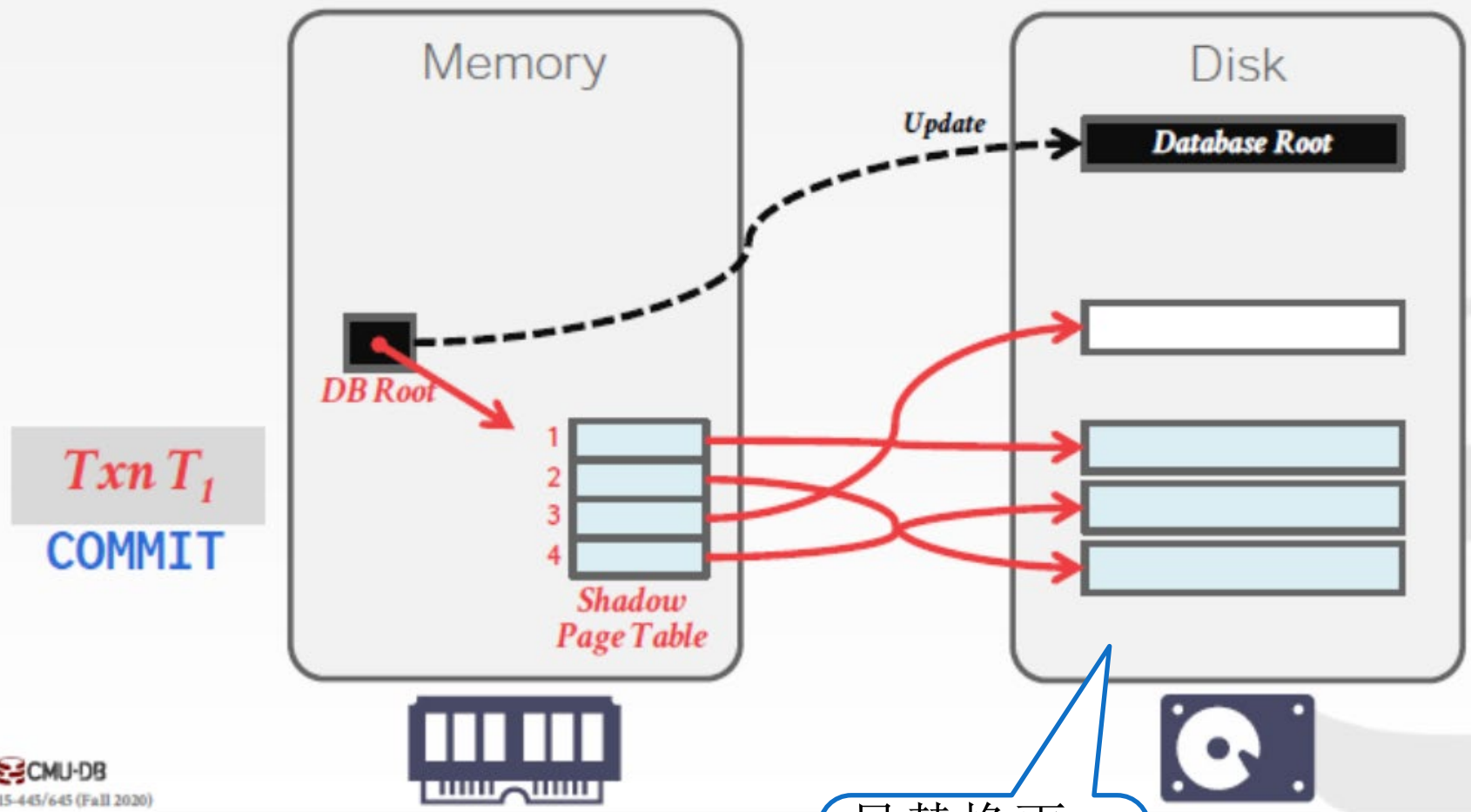
Active modifying txn
updates shadow pages.

Update

Disk

Database Root





6、日志管理

- 1) 按事务操作执行时间顺序记日志（多个事务操作并发）；
- 2) 须先写日志后写**DB**文件！！！！！



先写日志协议——WAL（ Write Ahead Logging）

7、日志的用途

- 1) 事务恢复
- 2) **DB**故障恢复
- 3) 系统分析



联机日志文件和归档日志文件

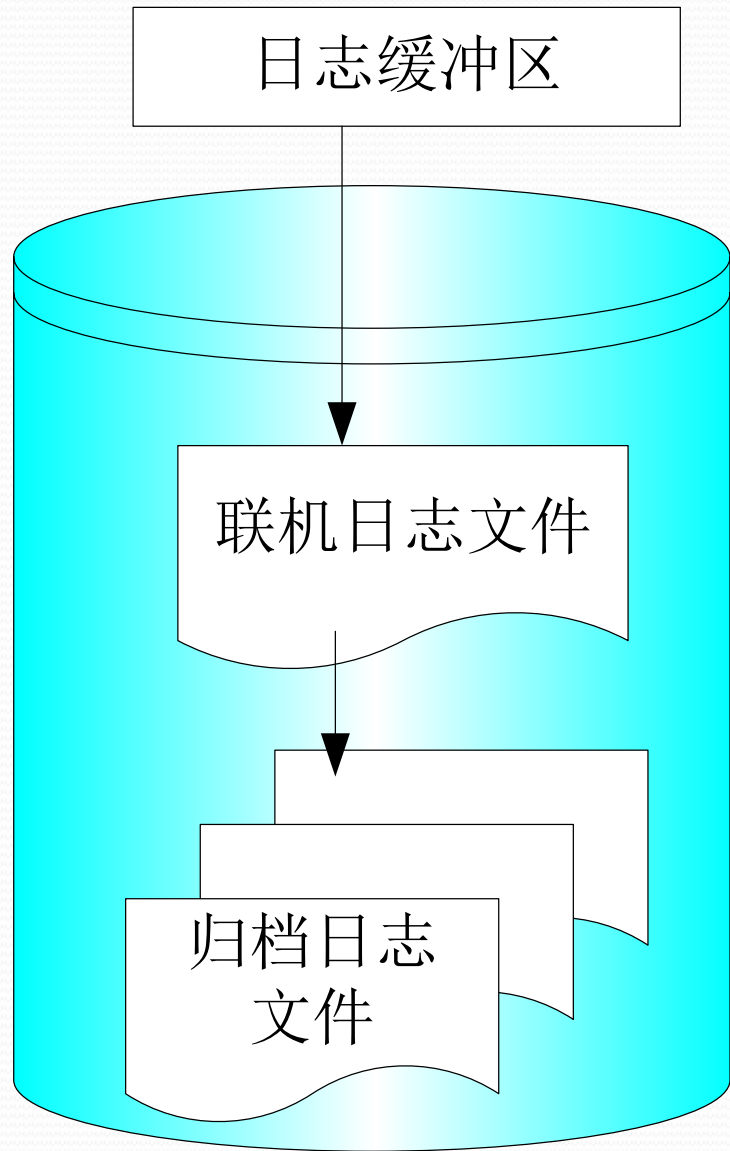
日志文件可分为两类：

1) 联机日志文件

最新日志内容，但大小有限，直接和**DBMS**日志缓冲区关联，保存数据库**当前一段时间**内的事务执行的变化过程，主要用于**事务故障**和**系统故障**。

2) 归档日志文件。

保存历史上所有的**不再用于联机处理的**日志记录，由联机日志文件归档而成，主要用于**介质故障**的恢复。



日志记录的使用（操作）

REDO——可理解为将日志记录对应的操作执行一遍，运用数据的**后像**。

UNDO——可理解为将日志记录对应的操作的逆操作执行一遍，运用数据的**前像**。

注： 1) 执行的方向；
2) DB的状态。

■ 幂等性

Why?

每个日志记录的**UNDO**操作和**REDO**操作都具有幂等性，即无论重复执行多少次，效果等同于执行一次。



日志文件是一个单调递增的文件

每个日志记录在日志中都有一个唯一的码，叫做日志序号（Log Sequence Number，简称LSN）。

日志文件是按照**LSN单调递增的顺序文件**，如果操作A的日志记录在操作B的日志记录之后生成，则 $LSN(A) > LSN(B)$ 。

LSN的实现（地址的递增与逻辑序号递增一致的策略）

一般由日志文件序号和记录在文件中的相对地址两部分组成。

日志的写操作问题：事务**撤销如何遵守**日志文件的单调递增特性？

redo-only日志（补偿日志，compensation log）：Undo操作除了将数据项设置成旧值，还额外生成一条“redo-only”日志记录来体现该数据的undo更新，该日志记录不需要包含数据项的旧值。

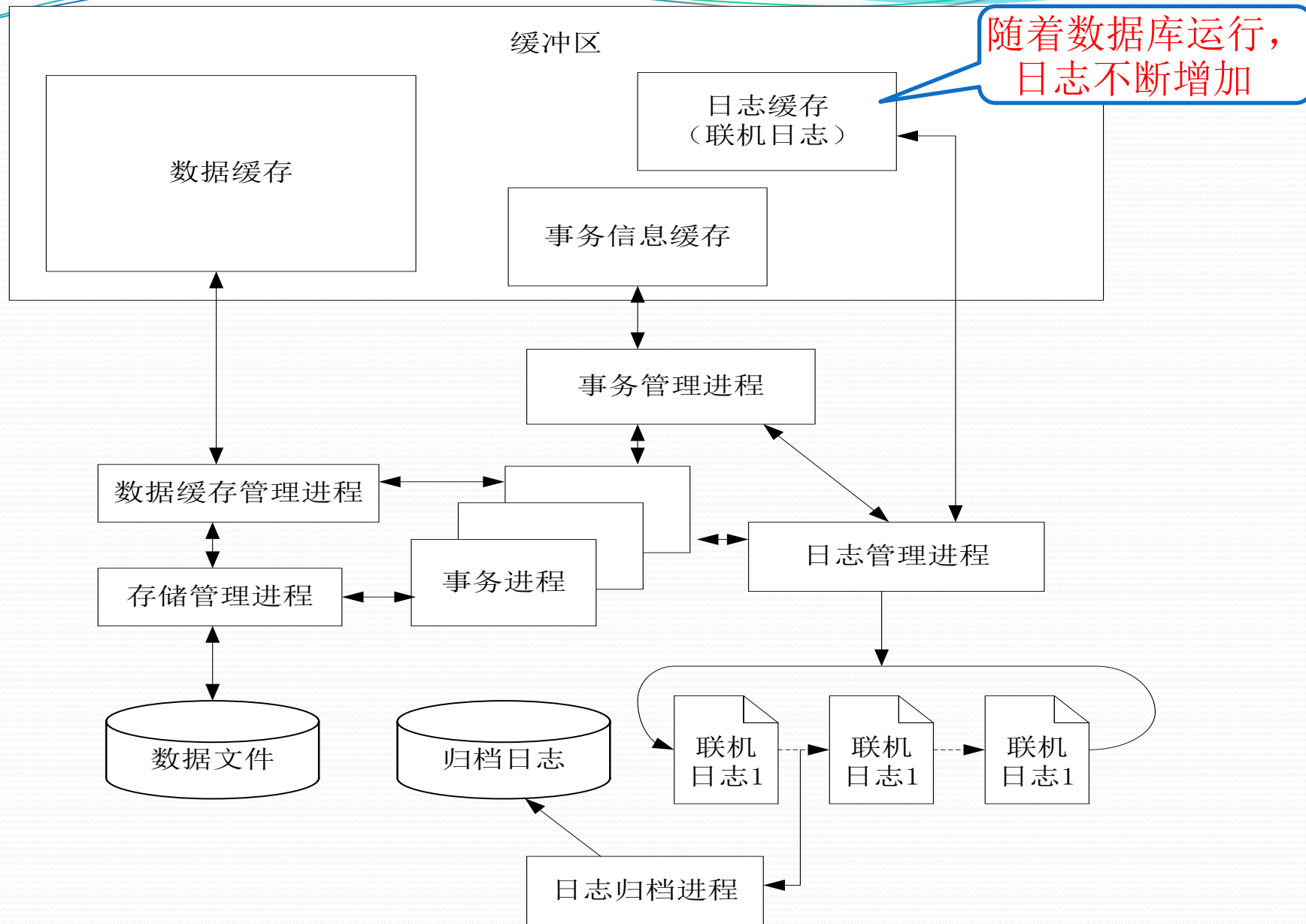


日志记录的REDO和UNDO执行方式

- 对日志文件中的多条日志记录进行REDO，要按照**日志序号(LSN)递增**的顺序进行，即正向扫描日志文件进行REDO。
- 对日志文件中的多条日志记录进行UNDO，要按照**日志序号(LSN)递减**的顺序进行，即反向扫描日志文件进行UNDO。

当要对日志文件中的多条按自然顺序交错排列的日志记录进行REDO和UNDO操作时，只需各自按照上述规则进行，原理上讲，REDO和UNDO之间**原则上没有先后的要求，执行多次也没有影响。**

现有系统中的ARIES经典恢复算法
中二者有处理上的先后顺序





缓存中的日志写出到磁盘日志文件的机制

缓存中的日志内容单调递增，缓存资源有限



DBMS需要建立缓存中的日志写出到磁盘日志文件的机制。

其中，触发日志缓存页面写出到外存联机日志文件的原因包括：

- 事务提交
- 缓冲区使用达到一定限度
- DBMS的关机shutdown、检查点



日志相关的协议（保证事务的原子性和持久性）：

WAL日志先写、提交、成组提交



先写日志协议WAL——在覆盖一个外存页面之前，必须先强制写出该页面新版本对应的日志记录。



WAL协议实现技术：每个数据页面有一个字段记录最近版本对应的日志记录**LSN**，写出该页面前，调用**Log_flush (LSN)**内部函数将该**LSN**之前的所有日志记录写出。

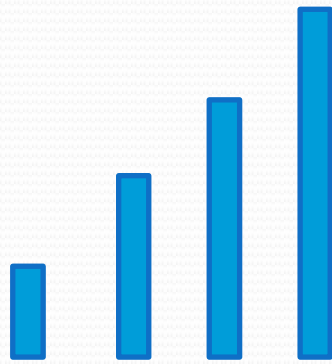
提交时强制写日志协议force-log-at-commit——作为提交工作的一部分，必须强制写出该事务的所有日志记录。

实现技术：调用**Log_flush (COMMIT_LSN)**将该事务的**commit**日志记录及其之前的所有日志记录写出。

可能迟滞

成组提交优化技术：当日志记录产生较为频繁时，凑齐一组（一个日志缓存页面）的日志记录才执行日志缓存写出操作，在此之前该组日志对应的事务提交均处于等待状态。

优化整体性能



Wait: 1 2 3 4
Avg (wait) 2.5



2 2 2 2
2



7、缓存管理

数据缓存中的页面可分为两类：

干净页面——由已提交的事务更新完的页面

脏页面——由未完成的事务更新的页面

上述两类页面带来的系统性能相关问题

(1) 干净页面的写出策略

强制写（force）——事务提交时**必须先写出**所有该事务相关的干净页面，然后才能完成提交。

非强制写（no-force）——即使事务修改的磁盘块**不必等到**全部写回到磁盘，也允许它提交完成。

■ 强制写与非强制写对性能的影响

强制写策略增加了**事务提交的响应时间**，还造成**频繁的I/O**操作，降低了系统的**事务吞吐率**。



(2) 脏页面写出的策略

隐 形 (steal) ——脏页面可**随时**由于页面替换被写出

非隐形 (no-steal) ——脏页面**被固定**在缓冲区中，不允许被替换出去，直到事务结束（提交或者夭折）

■ 隐形与非隐形对性能的影响

非隐形策略增加了对**缓冲区容量**的要求

相关实现机制： 缓存页面固定

原理： 对数据页面封锁

锁的类型： 闩锁 (latch)

缓冲页面上的闩锁与事务并发控制的锁无关，是一种短期持有的锁。



先写日志协议WAL要求缓冲页 B_x 的相关日志必须在 B_x 输出之前输出到磁盘，而为了保证该协议，在输出缓冲页 B_x 到磁盘时，不能允许同时有向 B_x 缓冲页的写操作。

缓冲页面相应措施：当事务要对一个数据项执行写操作时，需要先获得该数据项所在页面的排它锁（闷锁），并且更新完后立即释放该锁。

使用闷锁的缓冲页 B_x 输出到磁盘的过程：

- （1）获取 B_x 上的排他锁闷锁，以确保没有其他事务正在对 B_x 执行写操作；
- （2）将 B_x 相关的全部日志记录输出到磁盘；
- （3）将 B_x 输出到磁盘；
- （4）释放 B_x 上的排他锁闷锁。

申请和释放缓冲页面的闷锁即使不遵循两阶段锁协议，也不影响事务的可串行性。



强制写与非强制写对日志与恢复的影响

强制写策略减少了对日志记录中REDO信息的需求，恢复时一般不需要REDO操作。

隐形与非隐形对日志与恢复的影响

非隐形策略减少了对日志记录中UNDO信息的需求，恢复时一般不需要UNDO操作。



不需要REDO?

不需要UNDO?

强制写+非隐形策略? → 不需要日志?

日志仍然不能省略, 原因:

- 1) 介质故障→REDO信息的必要性
- 2) 提交的过程中, 页面是脏的, 隐含的采用了隐形策略, 除非采用复杂的影子页算法, 保证提交操作的原子性。

■ 因此, 一般采用非强制写+隐形策略, 以降低系统开销, 提高响应能力。



日志的开销大。



操作系统在缓冲区管理中的作用

有两种管理缓存的技术途径：

- (1) 为DBMS保留部分专属的主存，DBMS的子系统直接负责管理这部分缓存。
- (2) DBMS在操作系统提供的虚拟内存中实现缓冲区。目前多数操作系统会完全控制虚拟内存，会通过交换区（swap space）磁盘空间来保留不在主存的虚拟内存页。数据库文件和虚拟内存中的缓冲区之间的数据传输必须由DBMS管理，从而实现先写日志协议。

当DBMS要输出缓存页 B_x 时，操作系统先从交换区输入 B_x ，然后可能两次 B_x 输出（一次DBMS输出，一次操作系统输出）。

10.5 恢复策略

1、事务故障恢复

——因各种故障导致事务未执行完而**abort**时的恢复。

1) 目标：维护原子性

2) 恢复步骤

① 反向扫描日志文件：

——查事务执行过的更新操作；

② 执行该事务的最后一条日志记录的**UNDO**操作；

③ 循环执行上述操作并同样处理，直至事务开始标记。

3) 特点

DBMS自动完成



2、DB故障恢复

1) 系统故障

——撤消故障发生时**未完成事务**和重做**已完成事务**的恢复。



① 目标：持久性

注意两种操作的执行方向

② 步骤

[1] 正向扫描日志文件；

[2] 找出故障发生前**已提交**事务，该事务标识记入REDO队列；

[3] 找出故障发生时**未完成**事务，该事务标识记入UNDO队列；

[4] 依照日志记录**反向顺序**对UNDO队列中事务进行UNDO操作：

（反向扫描日志文件，执行该事务的UNDO操作）；

[5] 依照日志记录**正向顺序**对REDO队列中事务进行REDO操作；

（正向扫描日志文件，执行该事务的REDO操作）。

③ 特点：DBMS自动完成。

思考问题：查找
日志文件的范围

2) 介质故障

——数据和日志文件破坏时的恢复。

① 目标：持久性

② 方法

a: 向前恢复（恢复未写出的提交数据，forward）→

[1] 装入后备付本：

——先恢复到最近备份时的正确状态；

[2] 装入系统日志文件；

[3] 正向扫描日志文件；

[4] 利用日志文件后映像（该备份点以后做了哪些操作）执行 REDO。

（前一个付本+REDO）

b、向后恢复（撤销已写出的未提交数据，backward） ←

[1] 装入后备付本；

[2] 装入日志文件；

[3] 反向扫描日志文件；

[4] 利用日志文件中前映像排除对DB的改变。

（当前映像+UNDO）

c、重运行

[1] 装入最新正备付本；

[2] 重新运行最近一次备份以来的事务。

优点：简单：无日志，只需登记已执行事务。

缺点：时间长；

重运行事务执行顺序可能变化。



■ 日志模式的具体实现策略（记录/数据块→物理/逻辑/物理逻辑）

物理日志（Physical logging）：记录字节级的数据库变化，例如记录一个页面内的某个地址的数据，实现一般以页面为单位。

存在的问题：存储开销大。

逻辑日志（Logical logging）：记录高级别的事务操作（一条逻辑日志可能涉及多个页面上的多个元组的更新），比物理日志耗费的存储空间更小。

存在的问题：基于逻辑日志的恢复技术实现复杂，尤其是日志包含了并发事务时，涉及原子性、并发正确性等问题。

物理逻辑日志（Physiological logging）：物理和逻辑两种日志技术的混合策略。物理角度以页面为单位（日志记录仅针对单个数据页面，对应内容不跨页），逻辑角度限于页面内部（例如记录某个数据页内某些槽（slot）的字节级变化。

原子性可控，存储开销不大，在现有的DBMS中较多使用。



例: UPDATE foo SET val = XYZ WHERE id = 1;

页面为单位的操作序列

Physical

Logical

~~Physiological~~

```
<T1,  
Table=X,  
Page=99,  
Offset=4,  
Before=ABC,  
After=XYZ>  
  
<T1,  
Index=X_PKEY,  
Page=45,  
Offset=9,  
Key=(1,Record1)>
```

```
<T1,  
Query="UPDATE foo  
SET val=XYZ  
WHERE id=1">
```

```
<T1,  
Table=X,  
Page=99,  
Slot=1,  
Before=ABC,  
After=XYZ>  
  
<T1,  
Index=X_PKEY,  
IndexPage=45,  
Key=(1,Record1)>
```



abort日志：当回滚事务Ti的所有undo操作都完成后，系统为该事务写一个**<Ti abort>**日志记录，表明撤销完成了，也**对应事务的一种结束状态**。

该机制可使得每个事务的**undo过程至多完整执行一遍**。

引入**abort**日志后，当发生系统崩溃后，扫描日志文件，当发现**<Ti start>**日志记录时：

- 若未发现**<Ti commit>**，也没有**<Ti abort>**，则需要对该事务的所有日志记录执行撤销操作；
- 若发现**<Ti commit>**或者**<Ti abort>**日志记录，都标识事务到达结束状态，**都会对该事务的日志记录执行redo操作**。

abort日志让“夭折”变“提交”



B+树索引的undo操作

存在的问题：**层次索引**通常采用的并发控制策略会在获取下层结点的锁之后尽快释放祖先结点的锁（蟹行），因此，插入操作可能在**undo**之前已释放结点锁，从而出现其他事务已经读/写相关中间结点的内容。此时，若插入操作的**undo**直接用旧值代替结点的新值，则会出现并发错误。

解决思路：在B+树上的插入和删除操作**采用逻辑操作**（便于撤销），同时**使用逻辑和物理日志**。



索引更新过程:

- (1) 在执行修改索引的操作之前, 事务 T_i 创建一个 $\langle T_i, O_j, \text{operation-begin} \rangle$ 日志记录, O_j 为操作实例的唯一标识;
 - (2) 开始记录后, 操作所做的所有更新按正常方式创建更新日志记录, 对应物理日志记录;
 - (3) 操作结束后, 写入一个形如 $\langle T_i, O_j, \text{operation-end}, U \rangle$ 的日志记录, 其中 U 包含undo信息, 对应逻辑日志记录, 例如插入的undo信息为删除。
- 该过程中, 逻辑日志仅用于撤销, 不用于重做。



引入索引逻辑日志后的恢复动作原理：

新增加的“operation-end类型日志记录”标识已完成的逻辑操作，其回滚和其他操作不同。



一旦系统发现一个 $\langle T_i, O_j, \text{operation-end}, U \rangle$ 的日志记录，就使用其中的U信息执行逻辑undo，但是逻辑undo过程中依然会对产生的内容更新生成物理undo日志（补偿日志）。

执行完逻辑undo后，写入一条 $\langle T_i, O_j, \text{operation-abort} \rangle$ 日志记录，表示索引操作回滚完成。之后，恢复动作直接跳过 T_i 之前的日志记录，直至遇到 $\langle T_i, O_j, \text{operation-begin} \rangle$ 日志记录。



引入索引逻辑日志后的恢复动作原理（续）：

常规恢复时，如果遇到一个 $\langle T_i, O_j, \text{operation-abort} \rangle$ 日志记录，则跳过 T_i 中 O_j 前面所有的记录（包括 O_j 的 operation-end 记录），直至 $\langle T_i, O_j, \text{operation-begin} \rangle$ 日志记录。

直至遇到 $\langle T_i, \text{start} \rangle$ 日志记录时，事务回滚完成，向日志中写入一个 $\langle T_i, \text{abort} \rangle$ 日志记录。

10.6 具有检查点的恢复技术

10.6.1 产生的原因

利用日志恢复的过程需要扫描全部的日志记录，进行相关的恢复操作，而日志文件过大将带来大量的恢复操作。

恢复涉及两类操作，redo和undo，二者都是“必须”的么？

很多需要redo的事务的更新操作结果可能已经被写入磁盘文件中，对其redo没有必要。

↓ *问题：是否已写出存在不确定性，
解决思路：增加确定性*

优化机制——周期性的“确定”：建立检查点（checkpoint）。

10.6.2 检查点机制

在日志中增加**新的一类记录**（检查点记录），并**增设重新开始文件**。

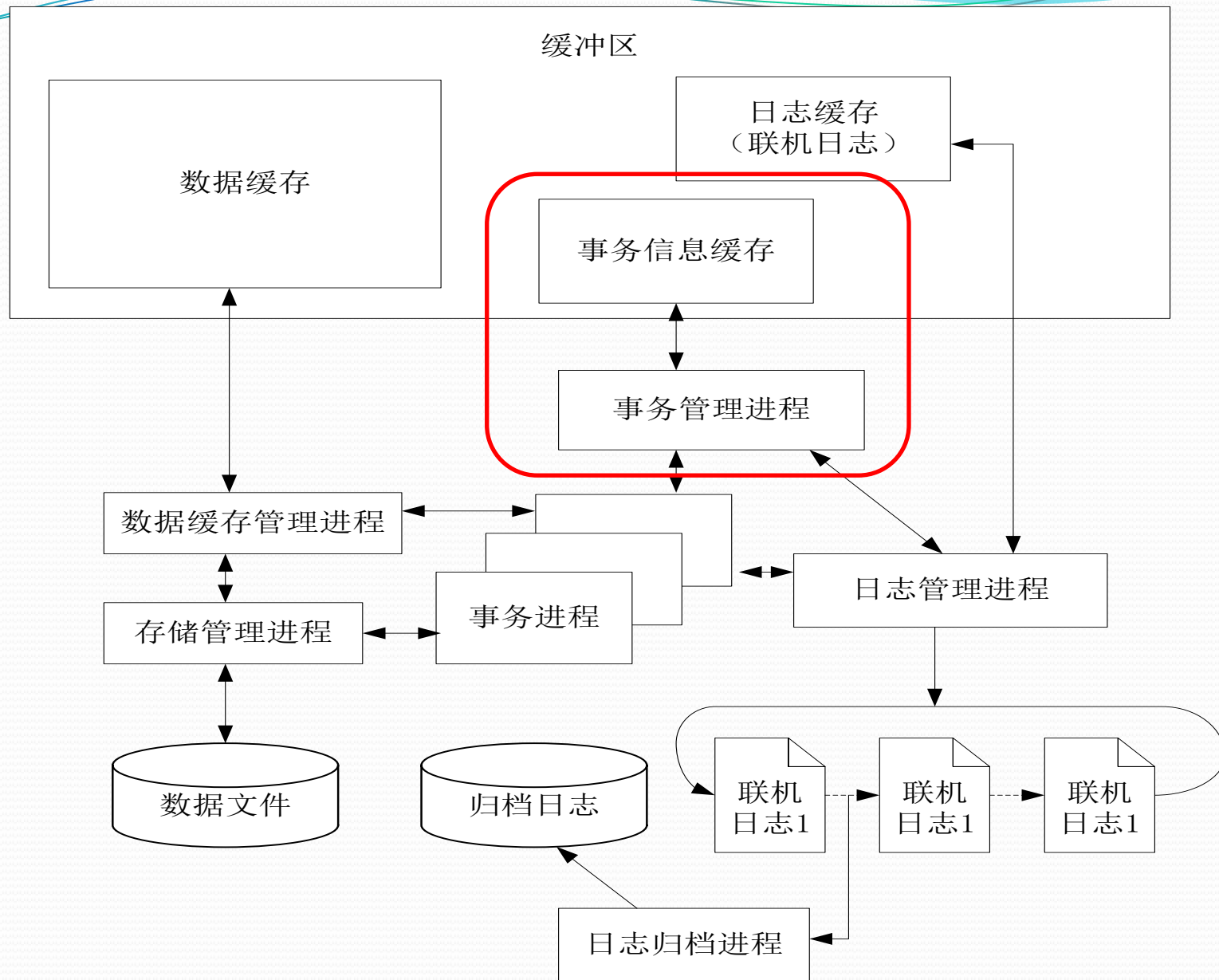
•检查点记录内容：

- 1) 建立检查点时刻所有正在执行的事务清单（Active Transaction Table）；
- 2) **上述事务最近一个日志记录的地址。**



目的？

【注】：检查点技术→日志归档的可行性





生成检查点的时机——周期性

时间周期

日志记录周期

- 检查点的动作（基本检查点策略，清晰检查点）

建立检查点，保存数据库状态。

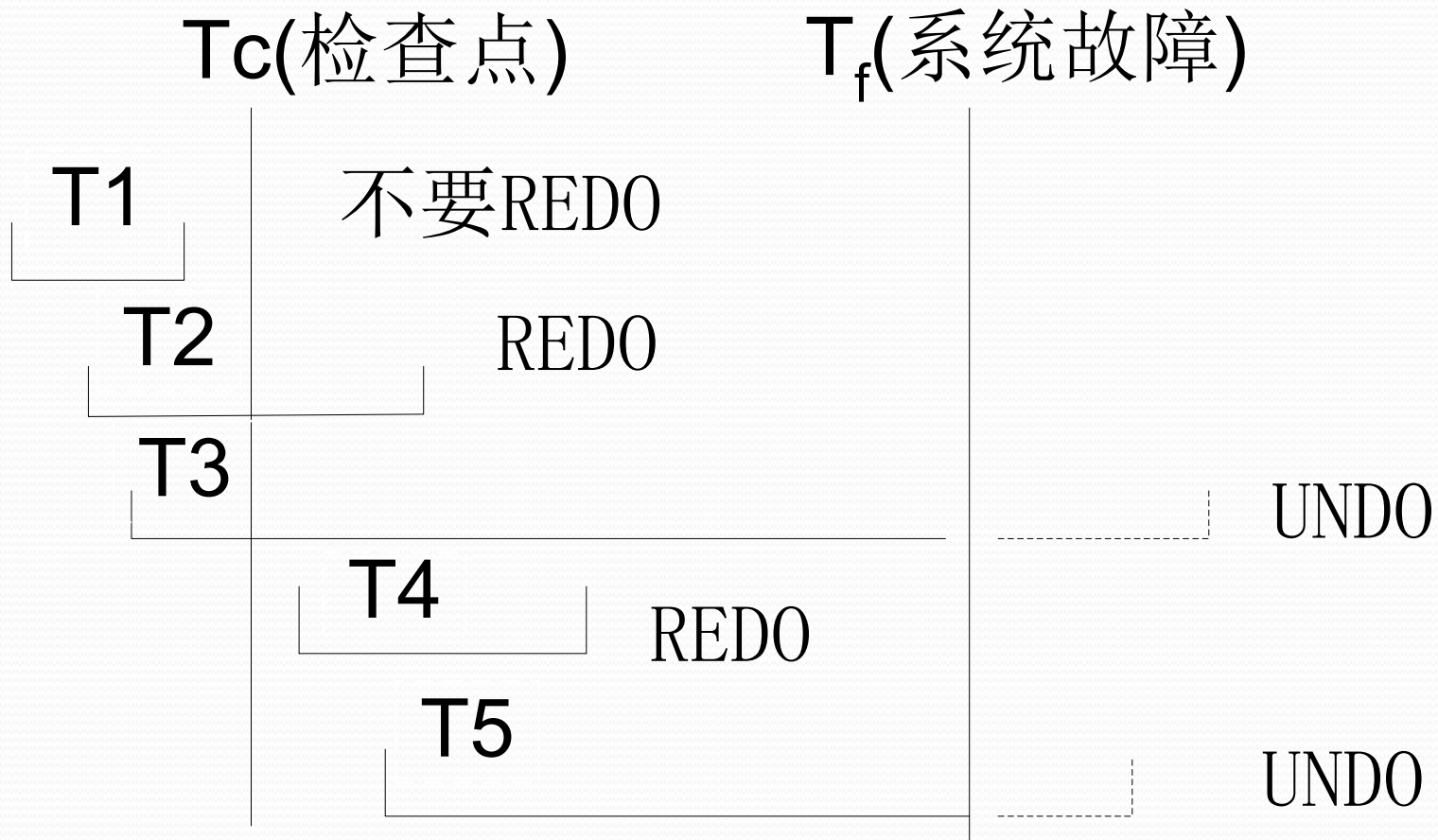
- 1) 将当前日志缓冲区中的所有日志记录写入磁盘的日志文件；
- 2) 在日志文件中写入一个检查点记录；
- 3) 把当前数据缓冲区的所有数据记录写入磁盘数据文件；
- 4) 在重新开始文件中记录检查点记录的地址。



缓冲页面的闕锁可以在检查点过程中保护缓冲页面不更新，从而没有产生新的日志记录，检查点执行完成后释放闕锁。

•使用检查点的恢复技术

- 1) 从重新开始文件中找到最后一个检查点的信息，并从日志文件中找到该检查点记录；
- 2) 从检查点记录中得到**ACTIVE-TRANSACTION-LIST**，并暂时将其全部列入**UNDO-LIST**队列，而**REDO-LIST**队列初始化为空；
- 3) 从检查点开始正向扫描日志文件，新开始的事务并入**UNDO-LIST**，遇到事务提交的日志记录，则该事务从**UNDO-LIST**移入**REDO-LIST**，直到日志文件尾；
- 4) 以检查点记载的最早日志记录和日志文件末尾为界，分别对**UNDO-LIST**和**REDO-LIST**执行**UNDO**和**REDO**操作。

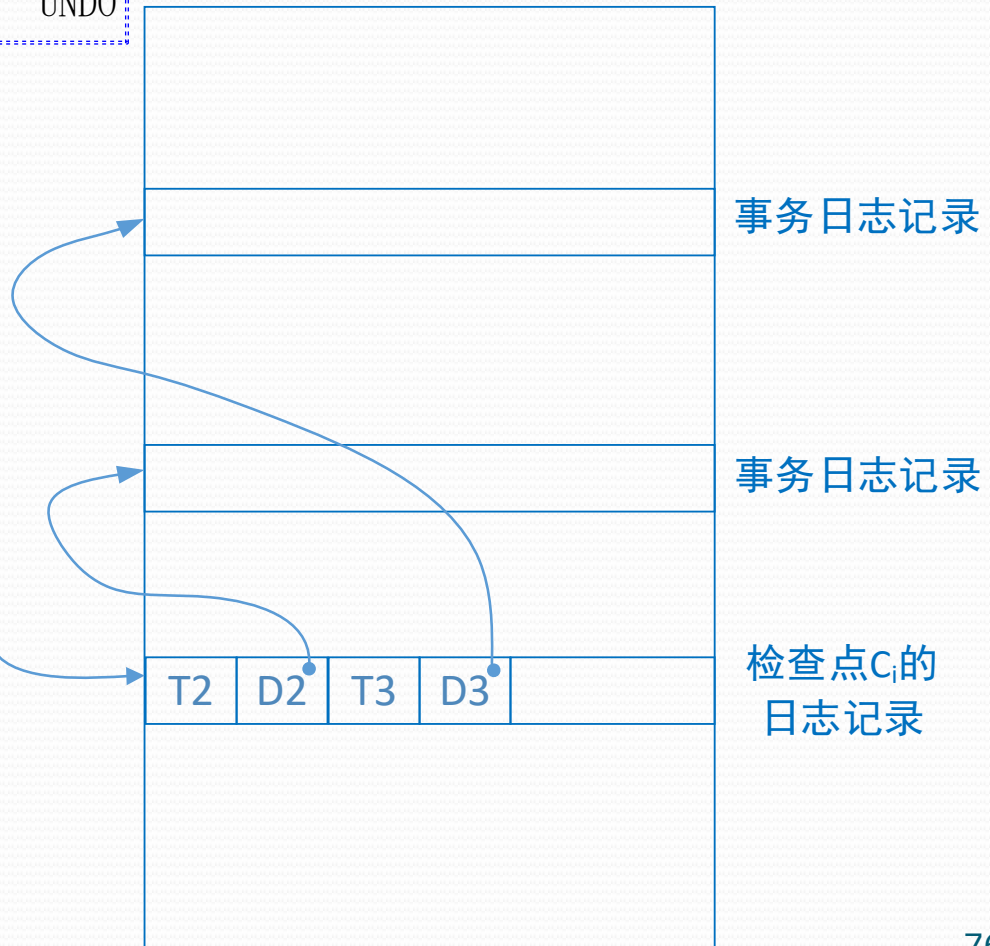


T _c (检查点)	T _f (系统故障)
T1	不要REDO
T2	REDO
T3	UNDO
T4	REDO
T5	UNDO

检查点C_i的
重新开始记录

C_i检查点记录地址

重新开始文件





原始的阻塞检查点

在较为原始版本的基本检查点策略中，检查点进行过程中需要暂停对缓冲区页面的更新（可通过闕锁实现），类似快照，禁止开启新事务，同时等待活跃事务结束，会导致系统的停顿。如果缓冲区页面数量很大，则清晰检查点的时间会很长，导致对事务处理的中断影响较大。

引入概念

脏页面列表（Dirty Page Table, DPT）：缓冲区中所有更新过的页面列表（含未提交事务修改过的页面），每个脏页面中有信息项来记录最开始导致该页面为脏的事务日志记录的LSN（recLSN）。



阻塞检查点改进思路：

在检查点记录写入日志后、修改过的缓存页写到磁盘之前，仍然不能开启新事务，同时阻塞当前事务，而不用等待当前活跃事务结束。

基本改进策略：

（1）将最后一个检查点记录在日志中的位置存在磁盘上专用的位置last_checkpoint;

（2）在写检查点记录之前，创建所有修改过的缓冲页列表DPT，仅当该列表中的所有缓冲页都输出到磁盘后，才更新磁盘上的last_checkpoint信息。



进一步改进：模糊检查点

执行检查点过程中可以开启新事务，活跃事务还可以更新数据。

在日志中增加新的日志记录来描述检查点的边界：

- <CHECKPOINT-BEGIN>：记录检查点的开始；
- <CHECKPOINT-END>：记录检查点的完整过程结束，其中包含活跃事务列表（ATT）和脏页表（DPT），而CHECKPOINT-BEGIN后开启的事务则不列入ATT列表。



转储（dump）的实现技术

➤ **归档转储（archival dump）**，可用于保留数据库的旧状态。

典型实现方法：在转储过程中不允许有活跃事务，执行过程类似检查点。

- （1）**日志缓存**内容写出到磁盘；
- （2）**数据缓存**页写出到磁盘；
- （3）将数据库的内容拷贝到**转储介质**；
- （4）将日志记录拷贝到**转储介质**。

其中第1、2、4步类似检查点动作。

可以有模糊转储机制，类似于模糊检查点策略，允许转储过程中事务仍然是活跃的。

➤ **SQL转储（SQL dump）**，将SQL DDL和SQL insert语句写到文件中，可以基于这样的文件重建数据库。在移植数据库（例如版本更新）时，数据库的物理位置、布局可能变化，此时**SQL转储**比较实用。

10.6.3 ARIES恢复算法

利用语义的恢复（遵守隔离性）算法，Algorithm for Recovery and Isolation Exploiting Semantics。

算法运行环境：缓存-日志及其相关数据结构

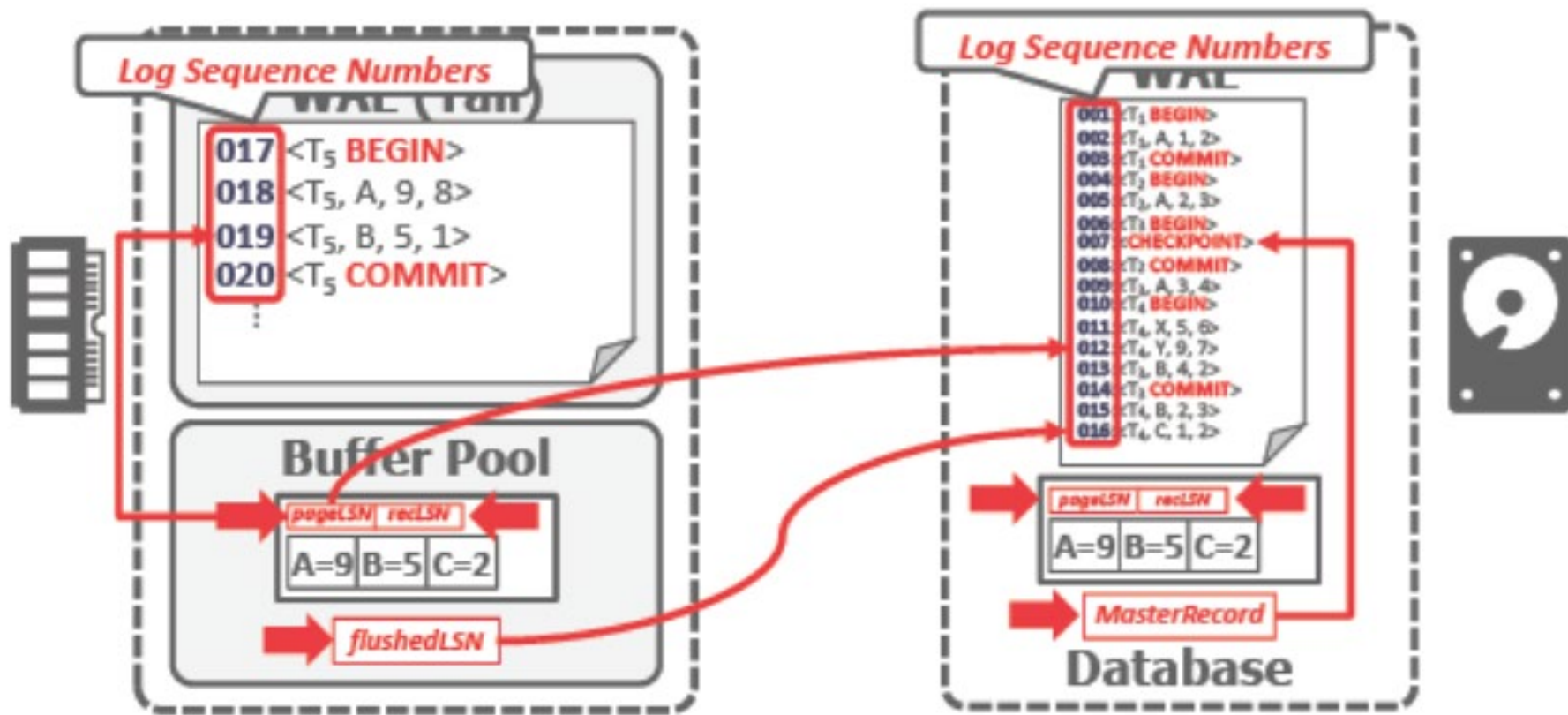
flushedLSN：缓存中记录的上一次写出的日志LSN号。

pageLSN：每个页面 $page_x$ 中记录的最新更新操作对应的日志LSN号。

recLSN：每个页面 $page_x$ 中记录的自从上一次从缓存中写出后的首次更新该页面的操作的日志LSN号。

lastLSN：缓存每个事务都有一个lastLSN，对应该事务的当前最新日志LSN号。

MasterRecord：缓存最新的检查点日志记录的日志LSN号。



TXN-END日志: 事务提交后，当commit日志被刷出到磁盘，DBMS返回一个事务提交的认可信息，之后某个时间点，DBMS会在日志中写一个TXN-END日志，用于系统内部提示，表示该事务后期不会再有任何日志信息了。（拓展、改变了abort日志的设计思想）

ARIES恢复算法执行过程

分为三个阶段：analysis、redo、undo

Analysis阶段

生成**ATT**（活动事务表）和**DPT**（脏页面表）信息。

- （1）读取检查点记录，初始化**ATT**和**DPT**，然后从检查点开始正向扫描日志；
- （2）当**遇见TXN-END**记录时，将该事务从**ATT**中移除；
- （3）当遇见其他事务记录（非commit记录）时，将事务加入**ATT**并标记状态为UNDO，当**预见commit**记录时，将该事务状态改为**COMMIT**；
- （4）当**遇见操作日志**记录时，若该页面P不在**DPT**，则将P加入**DPT**，并设置P的recLSN为该更新日志记录。

ARIES恢复算法执行过程（续）

Redo阶段

该阶段原则上**执行所有日志的更新操作**，即使是夭折事务的更新日志，同理，**也重做补偿日志（CLR）**。



系统从DPT页面中的最小recLSN对应的日志记录开始正向扫描日志文件，对于每一个遇到的日志记录或补偿日志记录，**除了特殊情况外**，均对该日志应用redo操作（包含刷新页面的pageLSN动作）。

存在跳过redo操作的特殊情况

Redo阶段的最后，为每个具有**COMMIT**状态的事务写入一个**TXN-END**日志记录，并将这些事务从**ATT**中移除。

ARIES恢复算法执行过程（续）

Redo阶段日志记录**跳过redo操作的特殊情况：**

- （1）该日志记录对应页面不在DPT中（检查点时清理过了）；
- （2）该日志记录对应页面在DPT中，但是日志的LSN小于该页的recLSN（之前的状态，已经应用于磁盘）；
- （3）该日志记录对应页面的pageLSN大于日志记录的LSN（幂等性）。

ARIES恢复算法执行过程（续）

Undo阶段

反向扫描日志文件，对undo列表（ATT中具有undo状态的所有事务）中的所有事务的日志记录执行撤销操作，同时撤销操作产生补偿日志（CLR），并且补偿日志的UndoNextLSN设置为该更新日志记录的PrevLSN值（跳过被补偿的操作）。

DBMS的启动意味着什么？

- 1) 先检查各个外存文件是否完好、正常，若发现问题，则需要人工进行介质故障恢复；
- 2) 查看联机日志，是否存在未提交事务，如存在则进行类似于系统故障恢复的处理。

好处：增强可靠性（不能保证DBMS每次都是正常SHUTDOWN）。

10.7 DB镜像 (DB mirror)

1、原因：介质故障：中断运行，周期备份，恢复麻烦

2、方法：利用自动复制技术（例如日志文件镜像）

3、策略

1) 整个**DB**或者关键数据复制到另一个介质（镜像磁盘）；

2) **DB**更新时，**DBMS**自动将更新结果复制到该副本；

3) 故障发生时，利用该镜像磁盘进行恢复。

4、优点

- 1) 无需关闭系统（自动进行镜像复制）；
- 2) 无需重装副本，自动保证一致性；
- 3) 提高可用性；
- 4) 提高并发性。

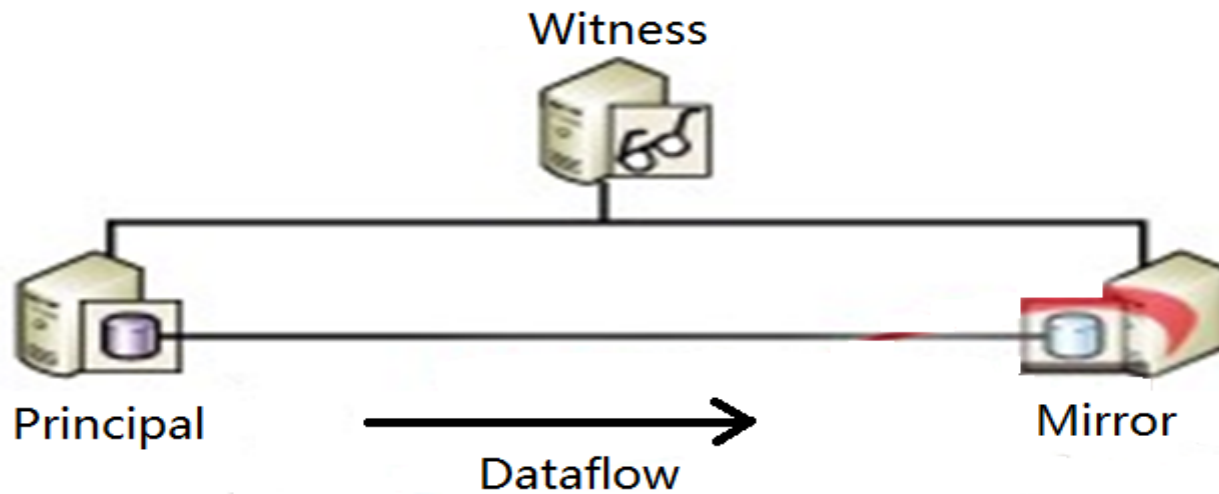
5、缺点

频繁复制更新，效率下降。

实用案例：双机热备（双机互备援Dual Active、双机热备份Hot Standby）、远程备份。

6、关键技术

如何监测（心跳线、PARTNER TIMEOUT、证人机制）、是否自动恢复。



基本+镜像+“证人”。三个服务器不断地ping对方，形成保留仲裁。如果服务器不能用，则其他服务器将确定怎样解决故障转移。考虑到机器所处的位置和网络的可靠性，主体服务器将会断开连接，见证服务器和镜像服务器仍然保留仲裁。

证人数据库是第三个SQL Server运行实例，当一个服务不可达并因此需要进行自动错误恢复的时候，证人服务器实现了2比1投票的能力。

例：假设日志记录有如下几种类型：

<START T>：表示事务T开始； <COMMIT T>：表示事务T已经提交；

<T,X,u,v>：表示事务T修改了数据X，其原来的值是u,更新后的值是v。

若某系统故障发生时，磁盘上日志文件的内容为：

<START T>;<T,C,29,30> <T,A,10,11>; <START U>;
 <U,B,20,21>;<T,C,30,31>;<U,D,40,41>;
<U,B,21,22> <COMMIT U>，请简述恢复的过程。

例：若某系统故障发生时，磁盘上日志文件的内容为：

<START T>;<T,C,29,30> <T,A,10,11>; <START U>;
<U,B,20,21>;<T,C,30,31>;<U,D,40,41>;
<U,B,21,22> <COMMIT U>，如何进行恢复？

恢复过程：

- (1)正向扫描日志，由于事务U有START和COMMIT，放到REDO队列中，而事务T只有START，故而放到UNDO队列。
- (2)对UNDO队列中进行UNDO操作，即对T进行UNDO操作，即反向扫描T的日志，即对C写30，对A写10，对C写29。
- (3)对REDO队列中进行REDO操作，即对U进行REDO操作，即正向扫描U的日志，即对B写21，对D写41，对B写22。

注意：写成类似“C由29变做30”是错误的。

课后作业第2题，
日志表述方式不同

思考：如果增加检查点记录，本题有何变化？

慕课讨论题

- 数据库系统哪些情况下会将缓存中的日志文件写出到磁盘？

日志在数据库系统的恢复中发挥着重要的作用，日志在什么情况下需要写出到磁盘文件？

- 检查点机制对性能可能产生哪些影响？

数据库系统的检查点是其恢复子系统的一种周期性执行的机制，该机制对于数据库系统的性能有哪些影响？