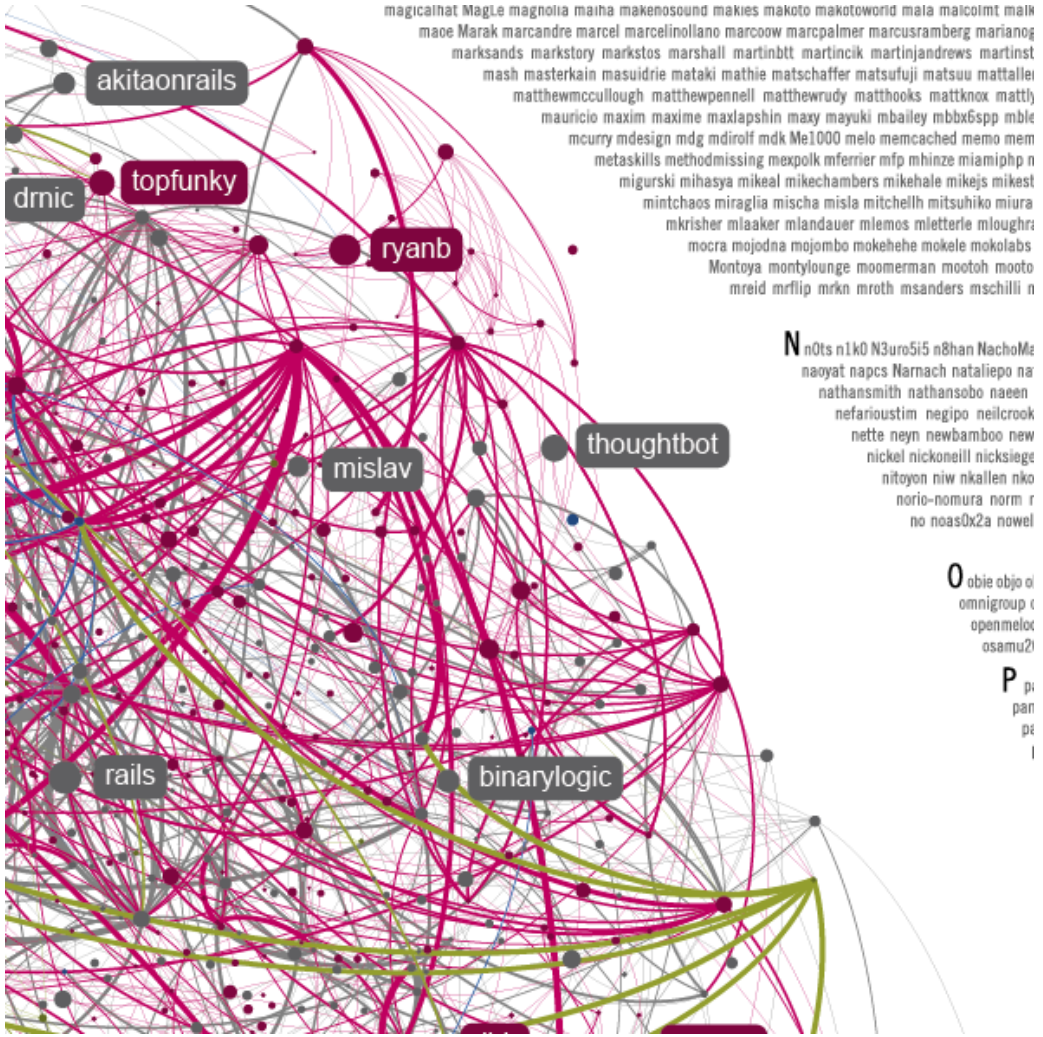
DATAGURU 炼数成金

DB2设计与性能优化

第3周



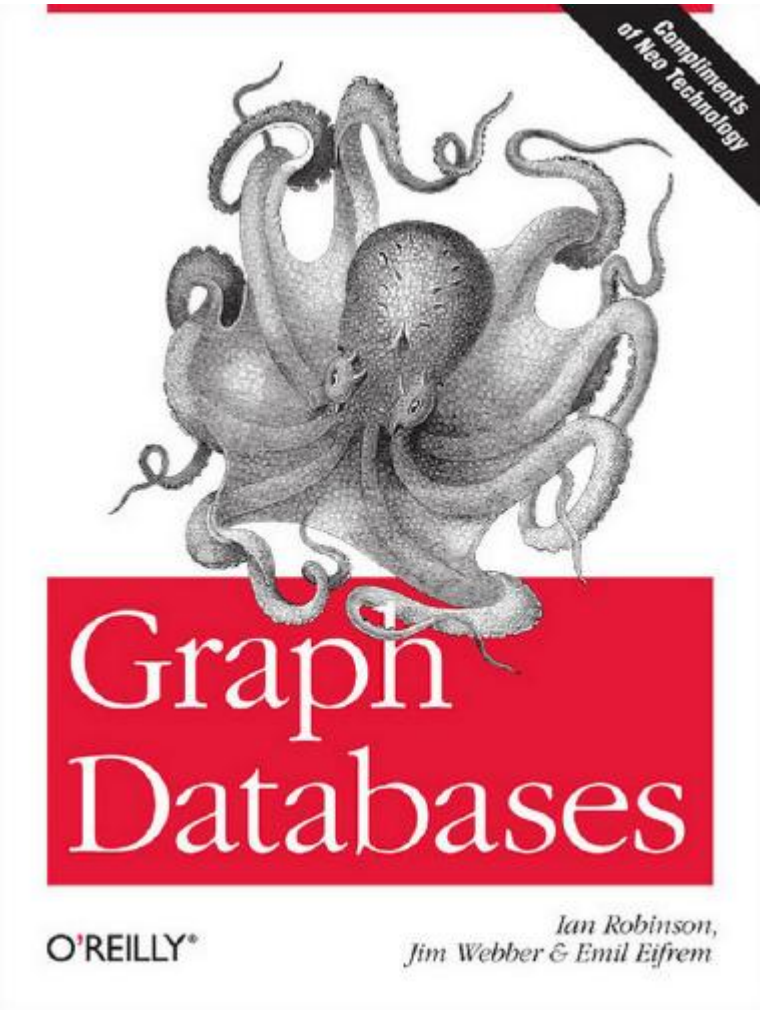
法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

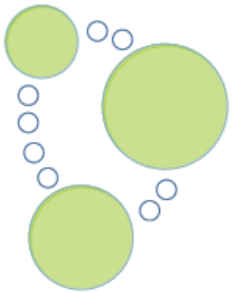
课程详情访问炼数成金培训网站

http://edu.dataguru.cn

图数据库参考书



The Neo4j Manual



neotechnology
value in relationships

本周内容

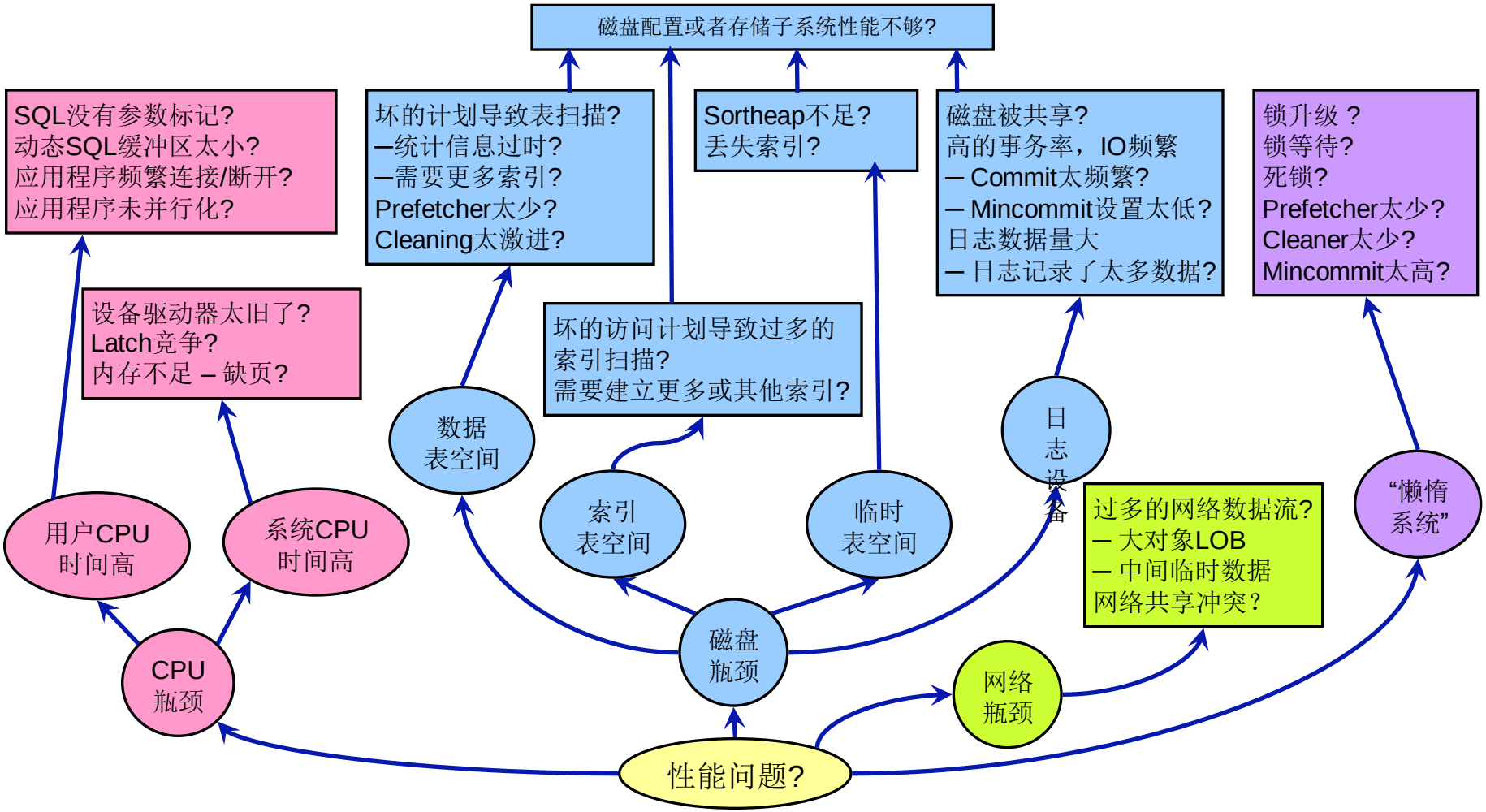
- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top

监控—飞机仪器室

DB2性能监控方法学

- 为什么监控？
  - 保证日常运行
  - 分析性能瓶颈，进行性能调优
- 如何监控？
  - 操作系统层面
    - IO、CPU、内存、网络等是否存在瓶颈
    - 系统工具：iostat, top, vmstat, netstat, sar
  - 数据库层面
    - 缓存池命中率，锁，SQL语句，事务吞吐，排序，日志等
    - 监控工具：快照、事件监控器、管理视图、 db2pd、 db2top等
- 怎么分析监控数据？
  - 最大问题：监控数据易于获得，却难以分析
  - 怎么有针对性的抽取相关的监控数据进行分析
  - 如何分析并计算性能指标，判断性能问题
  - 根据性能指标，对系统进行调优

监控性能瓶颈 -- “万变不离其宗”

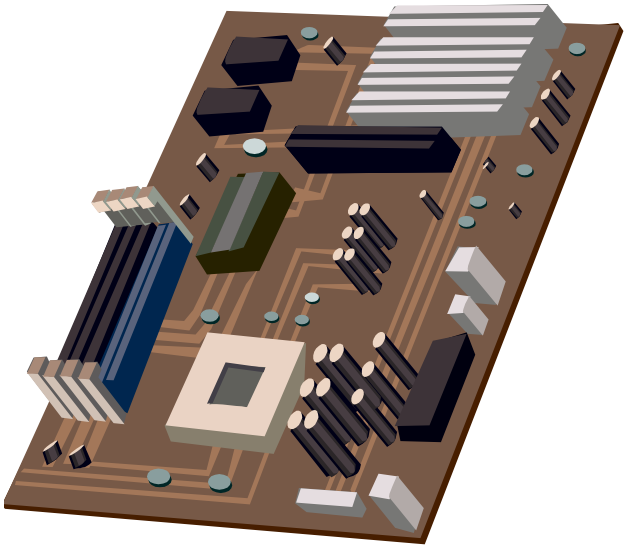


提纲

- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top
- 小结与练习

操作系统监控 — 从了解你的系统的开始

服务器所跑的操作系统及其版本?
多少个CPU处理器可用? CPU处理器的速度?
系统启动后有多少可用内存?
设置了多少交换空间?
当前存储设备的空间和性能?
服务器跑了哪些应用?



可用工具和命令

- db2pd –osinfo (All)
- systeminfo (Windows)
- free (Linux)
- bootinfo (AIX)
- yast – (Linux)
- smit – (AIX)
- lvm – (AIX, Linux)

操作系统监控—目标与工具

- 检查CPU命令
  - top; vmstat; sar –u; sar –q; tprof(AIX)
- 检查内存状态的命令
  - vmstat; free; sar –b; top; lsps(AIX); svmon(AIX)
- 检查I/O状态的命令
  - iostat; sar –d; sar –u; sar –b
- 检查网络状态的命令
  - netstat
- 检查进程情况的命令
  - ps; taskmgr.exe(仅windows)
- 其它有用的命令
  - lsof；ipcs; df; dd, nmon, topAS(AIX)

操作系统监控 — top（Linux实例）

监控 CPU，Mem，Swap和db2sysc主进程

Terminal

File Edit View Terminal Tabs Help

```
top - 11:11:16 up 12:08,  4 users,  load average: 2.04, 1.31, 0.66
Tasks: 108 total,   2 running, 106 sleeping,   0 stopped,   0 zombie
Cpu(s): 23.3%us, 22.4%sy,  0.0%ni,  0.5%id, 50.4%wa,  1.6%hi,  1.8%si,  0.0%st
Mem:   1036540k total,  1021740k used,    14800k free,   103844k buffers
Swap:  2096440k total,       20k used,  2096420k free,   437776k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
11152 db2inst1  25   0  399m 160m  99m S 29.2 15.8   1:28.85 db2sysc
 5739 root      14  -1 35476  11m 6272 S  3.6  1.2   2:08.87 X
19758 db2inst1  16   0 54256  17m 5500 S  1.4  1.7   0:00.14 db2
 8309 root      15   0  102m  16m  10m S  0.8  1.6   0:22.74 gnome-terminal
 5787 root      15   0 14648 9456 7748 S  0.6  0.9   0:06.58 metacity
19803 joe       21   0 44684  16m 5500 S  0.6  1.6   0:00.06 db2fm
 5824 root      15   0  102m  26m  18m S  0.4  2.7   1:06.01 nautilus
 5767 root      15   0 28992 9516 7740 S  0.3  0.9   0:13.70 gnome-settings-
 5822 root      15   0  5912 2436 1968 S  0.3  0.2   0:41.43 vmware-user
 5881 root      15   0 18484 5852 4132 S  0.3  0.6   0:06.58 gnome-power-man
19805 joe       25   0 35500  15m 4536 R  0.3  1.5   0:00.03 db2set
 3166 root      18   0 35428  16m 5316 S  0.2  1.6   1:38.63 db2fmcd
 5809 root      16   0 93076  13m  11m S  0.2  1.4   0:02.78 gnome-panel
 2974 root      15   0  2376  896  716 S  0.1  0.1   0:45.63 vmware-guestd
 5832 root      15   0 94220  15m  12m S  0.1  1.5   0:27.09 main-menu
 5843 root      15   0 46092 5452 4472 S  0.1  0.5   0:09.63 gnome-vfs-daemo
 8382 db2inst1  16   0  4412 1916 1428 S  0.1  0.2   0:01.62 bash
16952 db2inst1  16   0  2192 1040  784 R  0.1  0.1   0:02.53 top
    1 root      16   0   720  284  248 S  0.0  0.0   0:01.16 init
    2 root      34  19     0    0    0 S  0.0  0.0   0:00.03 ksoftirqd/0
    3 root      10  -5     0    0    0 S  0.0  0.0   0:01.06 events/0
```

操作系统监控 — iostat

- 系统是否 I/O 瓶颈?
- 用3-5秒的间隔运行iostat?
- 那块磁盘设备接近100% 忙 (% tm_act)?
- DB2的哪个表空间容器或者日志在使用这个磁盘?

Disks:	% tm_act	Kbps	tps	Kb_read	Kb_wrtn
hdisk6	28.0	73.8	4.2	8999945	3314136
hdisk7	98.7	44486.4	6.1	13790876	629660
hdisk8	0.0	0.6	0.0	90469	2176
hdisk9	0.8	74.4	3.1	9875281	2532636
hdisk10	0.5	54.6	1.7	9108228	3616
hdisk11	0.1	13.7	0.4	2279793	244

示例:列出占有最高读时间的表空间容器：

```
SELECT varchar(container_name,70) as container_name,
varchar(tbsp_name,20) as tbsp_name, pool_read_time
FROM TABLE(MON_GET_CONTAINER('',-2)) AS t
ORDER BY pool_read_time DESC
```

提纲

- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top

DB2监控工具多多

- 快照(Snapshot) → 最基本，全面
- 事件监视器(Event Monitor) → 专注
- 监视器管理视图(V9.7) → 轻量级、独特的SQL界面
- 健康监视器 → 系统健康
- db2pd → 强大，易读，开销小
- db2mtrk → 内存
- db2top → 文本交互界面
- 其他

DB2监控工具特点

工具名称	工具简介
快照	用以捕获有关数据库和所有已连接应用程序的信息
事件监视器	事件监视器返回有关CREATE EVENT MONITOR语句中指定的事件类型的信息，对于每个事件类型，将在特定时间点收集监视信息
监视器管理视图	从V9.7引入的轻量级系统监视替代工具，可以收集和查看系统、活动或数据对象的数据
健康监视器	使用一些健康指示器来评估实例和数据库性能的特定方面，并根据某个特定的阀值来发送警告
db2pd	用于从适当的DB2数据库系统内存设置中检索信息，并生成一份报告，此报告可用于监控数据库系统(或一个数据库系统的任意组件）和/或对其时行故障排除
db2mtrk	用于为实例、数据库和代理提供一份完整的内存状态报告
db2top	db2top结合来自所有数据库分区的DB2快照信息，使用基于文本的用户界面提供正在运行的DB2系统的动态实时视图(Windows上没有这个命令)

快照监控（snapshot）

- 快照
  - 最基本最常用的DB2性能诊断工具
  - 数据库活动某一时间点上监控数据
  - 监控的数据一般是一段时间的累计计数
  - 产生的数据量相对较小
  - 即时监控的实用工具
- 快照监控开关 – monitor switches
  - DBM参数：dft_mon_xxxx，默认情况下，监控应用程序继承此开关
  - 局部：每个监控应用程序的都有自己的开关，并可以打开或关闭
  - 全局：实例级，任意监控应用程序的开关打开，则实例的该监控开关也处于打开状态

监控器和计数重置

监控器组	监控器开关	DBM 配置参数	所提供的信息
缓冲池	BUFFERPOOL	dft_mon_bufferpool	缓冲池活动的数量（换言之，即所执行的读取和写入操作的数量，以及各次读/写操作所用时间）
锁	LOCK	dft_mon_lock	具有的锁数量，以及遇到的死锁循环数量。
排序	SORT	dft_mon_sort	所执行的排序操作数量、使用的堆数量、遇到的溢出数、排序性能。
SQL 语句	STATEMENT	dft_mon_stmt	SQL 语句处理开始时间、SQL 语句处理结束时间、SQL 语句标识。
表	TABLE	dft_mon_table	所执行的表活动数量，例如读取的行数、写入的行数等。
时间戳	TIMESTAMP	dft_mon_timestamp	时间和时间戳信息。
事务	UOW	dft_mon_uow	事务开始时间、事务结束时间以及事务的完成状态。

重置监控器计数值

```
db2  reset monitor all
db2  reset monitor for database sample
```

查看你的监控开关

db2 get monitor switches

```
C:\Documents and Settings\Administrator>db2 get monitor switches

           监视器记录开关

数据库分区号 0 的开关列表
缓冲池活动信息          (BUFFERPOOL) = OFF
锁定信息                    (LOCK) = OFF
排序信息                    (SORT) = OFF
SQL 语句信息              (STATEMENT) = OFF
表活动信息                 (TABLE) = OFF
获取时间戳记信息 (时间戳记)      = ON  2011-08-11 18:55:45.324652
工作单元信息                 (UOW) = OFF
```

当前监控会话中，LOCK的开关OFF

db2 get dbm monitor switches

```
C:\Documents and Settings\Administrator>db2 get dbm monitor switches

      收集到的 DBM 系统监视器信息

数据库分区号 0 的开关列表
缓冲池活动信息          (BUFFERPOOL) = OFF
锁定信息                    (LOCK) = ON  2011-08-11 23:25:09.303980
排序信息                    (SORT) = OFF
SQL 语句信息              (STATEMENT) = OFF
表活动信息                 (TABLE) = OFF
获取时间戳记信息 (时间戳记)      = ON  2011-08-11 18:55:45.324652
工作单元信息                 (UOW) = OFF
```

实例级LOCK开关ON，有其他会话打开了此开关

打开你的监控开关

```
db2  update  monitor switches using bufferpool on lock on sort on
db2  update  monitor switches using statement on table on uow on
db2  get dbm monitor switches
```

```
C:\Documents and Settings\Administrator>db2 get dbm monitor switches

      收集到的 DBM 系统监视器信息

数据库分区号 0 的开关列表
缓冲池活动信息                (BUFFERPOOL) = ON   2011-08-11 23:33:58.515496
锁定信息                      (LOCK) = ON   2011-08-11 23:25:09.303980
排序信息                      (SORT) = ON   2011-08-11 23:33:58.515531
SQL 语句信息                  (STATEMENT) = ON   2011-08-11 23:33:59.224457
表活动信息                    (TABLE) = ON   2011-08-11 23:33:59.224458
获取时间戳记信息（时间戳记）      = ON   2011-08-11 18:55:45.324652
工作单元信息                  (UOW) = ON   2011-08-11 23:33:59.224456
```

```
db2  update  dbm cfg using dft_mon_bufpool on dft_mon_lock on
db2  update  dbm cfg using dft_mon_sort on dft_mon_uow on
db2  update  dbm cfg using dft_mon_stmt on dft_mon_table on
db2  get dbm cfg | find /i "DFT_MON"
```

```
C:\>db2  get dbm cfg | find /i "DFT_MON"
   缓冲池                                (DFT_MON_BUFPOOL) = ON
   锁定                                  (DFT_MON_LOCK) = ON
   排序                                  (DFT_MON_SORT) = ON
   语句                                  (DFT_MON_STMT) = ON
   表                                  (DFT_MON_TABLE) = ON
   时间戳记                          (DFT_MON_TIMESTAMP) = ON
   工作单元                              (DFT_MON_UOW) = ON
```

收集更多监视器数据将增加数据库管理器的开销，这可能影响系统性能

快照—获取快照命令

- 数据库管理器（实例）快照
  - GET SNAPSHOT FOR DBM
- 应用快照
  - GET SNAPSHOT FOR applications ON *dbname*
  - GET SNAPSHOT FOR application applid <app id> ON *dbname*
  - GET SNAPSHOT FOR application agentid <agent id> ON *dbname*
- 数据库快照
  - GET SNAPSHOT FOR DB ON *dbname*
- 缓冲池快照
  - GET SNAPSHOT FOR bufferpools ON *dbname*
- 表快照
  - GET SNAPSHOT FOR tables ON *dbname*
- 锁快照
  - GET SNAPSHOT FOR locks ON *dbname*
- 动态SQL快照
  - GET SNAPSHOT FOR dynamic sql ON *dbname*
- 所有的快照数据
  - GET SNAPSHOT FOR all ON *dbname*

快照中的应用程序

```
 Database Snapshot

Database name                              = PRD
Database path                              = /database/prd/NODE0000/SQL00001/
Input database alias                       = PRD
Database status                            = Active
Catalog database partition number          = 0
Catalog network node name                  = db_svc
Operating system running at database server= AIX 64BIT
Location of the database                   = Local
First database connect timestamp           = 2008-03-30 17:39:43.625513
Last reset timestamp                       =
Last backup timestamp                      =
Snapshot timestamp                         = 2008-03-31 09:48:34.163227

Number of automatic storage paths          = 0
... ...
High water mark for connections            = 67
Application connects                       = 24347
Secondary connects total                   = 2062
Applications connected currently           = 67
Appls. executing in db manager currently   = 11
Agents associated with applications        = 131
Maximum agents associated with applications= 176
Maximum coordinating agents                = 63
```

- 系统运行的时间
- 负载– 当前系统运行的应用程序数目

缓冲池 & I/O

OLTP

```
                Database Snapshot
Buffer pool data logical reads               = 1801103
Buffer pool data physical reads              = 59768
Buffer pool temporary data logical reads     = 0
Buffer pool temporary data physical reads    = 0
Asynchronous pool data page reads            = 0
Buffer pool data writes                      = 44151
Asynchronous pool data page writes           = 43838
Buffer pool index logical reads              = 6908802
Buffer pool index physical reads             = 7813
Buffer pool temporary index logical reads    = 0
Buffer pool temporary index physical reads   = 0
Asynchronous pool index page reads           = 0
Buffer pool index writes                     = 1731
Asynchronous pool index page writes          = 1726
Total buffer pool read time (milliseconds)   = 80081
Total buffer pool write time (milliseconds)= 162749
Total elapsed asynchronous read time         = 0
Total elapsed asynchronous write time        = 157120
Asynchronous data read requests              = 0
Asynchronous index read requests             = 0
No victim buffers available                  = 33725
LSN Gap cleaner triggers                     = 108
Dirty page steal cleaner triggers            = 10
Dirty page threshold cleaner triggers        = 58
Time waited for prefetch (ms)                = 0
Unread prefetch pages                        = 0
Direct reads                                 = 48
Direct writes                                = 0
Direct read requests                         = 6
Direct write requests                        = 0
Direct reads elapsed time (ms)               = 0
Direct write elapsed time (ms)               = 0
Database files closed                        = 0
Data pages copied to extended storage        = 0
Index pages copied to extended storage       = 0
Data pages copied from extended storage      = 0
Index pages copied from extended storage     = 0
```

优先级:

- 高
- 中
- 低

命中率(Hit Ratio):
- 含义: 在BP读数据的%
- 计算: (LR-PR)/LR
- 优异: > 95%
- 良好: > 80%
- 示例: 数据: 97% 索引: 99%

页清除(page cleaning):
- 含义: cleaner异步完成写的%
- 计算: (async writes) / (total writes)
- 良好: > 95%
- 示例: 数据: 98% 索引: 98%

没有可用的牺牲缓冲区(No victim buffers available):
含义: 当需要干净页而找不到可用页的次数
良好: 低， 高的数字意味着cleaner太少； 一般比换出脏页次数更敏感

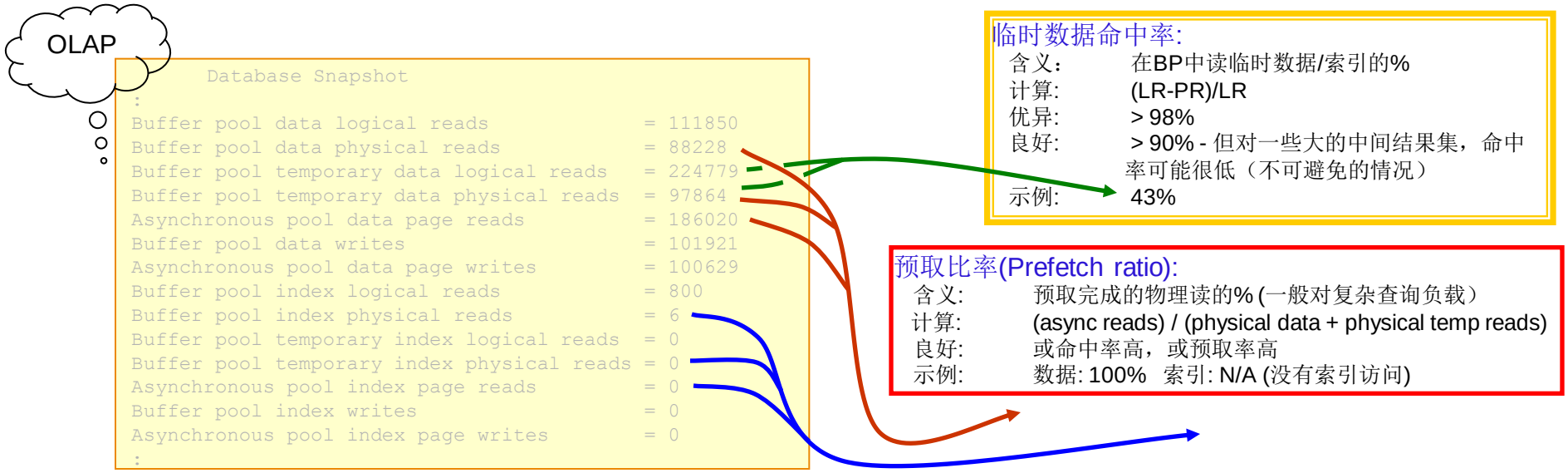
脏页换出次数(Dirty page steals):
- 含义: 读之前需将脏页换出的次数
- 良好: 非常低，对吞吐量有很大损害

Dirty page threshold / LSN Gap triggers
- 含义: 页清除器(page cleaners)触发次数，由于脏页太多或太旧
- 良好: 低，但比换出脏页次数影响小些

直接读/写次数 (Direct reads/writes):
- 含义: 不利用缓冲区的I/O次数 (longs/LOBs, catalog objs)
- 良好: 一般来说低比较好
- 示例: 动态SQL编译导致的一些直接读I/O

文件关闭次数(Files closed):
- 含义: 文件关闭/重打开的次数
- 良好: 当然低更好些，但是需要很高的值才对性能有影响

缓冲区与I/O(续)



- “数据库快照”中的缓冲区信息比“缓冲区快照”更完整更全面。
- 大部分的指标与工作负载类型无关—除了cleaning对DSS(OLAP)显得不如OLTP中重要
- DSS(OLAP)的命中率一般比OLTP的要低
- 时间相关的项是相对的，可用于分析“副作用”，比如设备利用率高导致的I/O太慢

读数据行的效率

```
              Database Snapshot
:
Rows deleted                = 5410
Rows inserted               = 78307
Rows updated                = 136903
Rows selected               = 206931
Rows read                   = 381967
:
```

Rows read vs. rows selected:

含义:	表示为找到目标的数据行需要读的行数
计算:	(rows read) / (rows selected)
良好:	≤ 2 or 3 for OLTP
示例:	1.85

- 显示索引效率的最佳信息，是否建立了有效的索引且索引被使用
- 不需要任何监控开关打开就会收集
- 在“表快照”中会进一步显示每个表读取的行数
- 行操作的数目和SQL语句的数目调试应用程序的极好高层次工具
  - 比如，你在一条SQL中或者一个表上所作的操作量是否如你预期？

动态SQL

```
              Dynamic SQL Snapshot Result
Number of executions                = 81341
Number of compilations              = 0
Worst preparation time (ms)         = 12
Best preparation time (ms)          = 12
Internal rows deleted               = 0
Internal rows inserted              = 0
Rows read                           = 81336
Internal rows updated               = 0
Rows written                        = 81336
Statement sorts                     = 0
Statement sort overflows            = 0
Total sort time                     = 0
Buffer pool data logical reads      = 162672
Buffer pool data physical reads     = 0
Buffer pool temporary data logical reads    = 0
Buffer pool temporary data physical reads   = 0
Buffer pool index logical reads     = 162732
Buffer pool index physical reads    = 0
Buffer pool temporary index logical reads   = 0
Buffer pool temporary index physical reads  = 0
Buffer pool xda logical reads       = 0
Buffer pool xda physical reads      = 0
Buffer pool temporary xda logical reads     = 0
Buffer pool temporary xda physical reads    = 0
Total execution time (sec.ms)       = 18.542262
Total user cpu time (sec.ms)        = 7.900000
Total system cpu time (sec.ms)      = 0.470000
Total statistic fabrication time (milliseconds) = 0
Total synchronous runstats time  (milliseconds) = 0
Statement text                      = Update DISTRICT …
```

语句编译:
- 含义: 执行前需要重新编译的次数
- 计算: (number of compilations) / (number of executions)
- 良好: 0，或者接近于0；调优措施：PCKCACHESZ太小?

读取行数:
- 含义:
  - 读的行数，也不是选择的行数
  - 在index-only access可能为0
  - 如果远远大于执行次数，可能意味着索引建的不好

语句的缓冲池命中率:
- 含义:
  - 含义跟BP或者DB层次的数据一样
  - 定位该SQL是否导致缓冲池问题的最佳方式

总时间:
- 含义: -该语句在数据执行的总时间

■对大多数语句来说应该关注的指标
- 执行次数，读写行数，执行时间, 排序等
- 关注和调优这些指标最大的语句，达到事半功倍的效果

排序

```
                Database Snapshot
:
Total Private Sort heap allocated          = 2058
Total Shared Sort heap allocated           = 0
Shared Sort heap high water mark           = 0
Total sorts                                = 24
Total sort time (ms)                       = 866719
Sort overflows                             = 15
Active sorts                               = 3
:
```

```
    Database Manager Snapshot
:
Private Sort heap allocated          = 2058
Private Sort heap high water mark    = 4101
Post threshold sorts                 = 0
Piped sorts requested                = 9
Piped sorts accepted                 = 9
:
```

排序溢出:
- 含义: 排序溢出写到磁盘的比率
- 计算: (sort overflows) / (total sorts)
- 良好: 对OLTP负载，0或接近于0

平均排序时间:
- 含义: 排序的平均运行时间
- 计算: (total sort time) / (total sorts)
- 良好: 远远小于语句预期的响应时间

排序内存超出:
- 含义: 排序堆高水位大于SHEAPTHRES值
  - 引发降低私有排序堆分配或可能溢出到磁盘

- 排序溢出的代价很高
  - 需要在调优时平衡SORTHEAP和bufferpool的分配
  - 如果排序总时间不大，那么不必太在意排序溢出
  - 调优能减少排序溢出，但可能很难除去最后的几个溢出(特殊的查询)
- 排序的监控数据也在“动态SQL快照”中体现

监控常见问题

- 执行了相应的快照，但是没有返回任何信息？
  - 通常是由于你没有打开相应的监控开关
- 监控对高负载系统性能有影响，避免打开所有的监控开关
- 使用监控时需要注意的什么？
  - 对于极大事务量的系统，在使用事件监控时，需要调整以下参数为合适的值，以避免系统停止响应
  - db2set DB2_MAX_INACT_STMTS=1000
  - db2 update dbm cfg using MON_HEAP_SZ xxxx

提纲

- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top

管理视图（V9.7）

视图名称	描述
SYSIBMADM.ADMINTABCOMPRESSINFO	表压缩信息
SYSIBMADM.ADMINTEMPTABLES	临时表信息
SYSIBMADM.APPLICATIONS	数据库中运行的应用
SYSIBMADM.APPL_PERFORMANCE	每个应用中rows selected与rows read的比率
SYSIBMADM.BP_HITRATIO	缓冲池命中率
SYSIBMADM.BP_READ_IO	缓冲池读的信息
SYSIBMADM.BP_WRITE_IO	缓冲池写的信息
SYSIBMADM.CONTAINER_UTILIZATION	表空间容器的利用率信息
SYSIBMADM.LOCKS_HELD	当前获得的锁信息
SYSIBMADM.LOCKWAITS	锁等待的信息
SYSIBMADM.LOG_UTILIZATION	日志利用率的信息
SYSIBMADM.LONG_RUNNING_SQL	执行时间最长的SQL
SYSIBMADM.SNAPAGENT_MEMORY_POOL	代理的内存使用情况
SYSIBMADM.SNAPBP	缓冲池基本信息
SYSIBMADM.SNAPDYN_SQL	数据库中动态SQL的执行情况
SYSIBMADM.SNAPLOCKWAIT	锁等待的情况
SYSIBMADM.SNAPSTMT	SQL语句的执行情况
SYSIBMADM.SNAPTAB	表信息
SYSIBMADM.SNAPTAB_REORG	重组信息
SYSIBMADM.SNAPTBSP	表空间信息
SYSIBMADM.TBSP_UTILIZATION	表空间利用情况
SYSIBMADM.TOP_DYNAMIC_SQL	消耗资源最多的SQL语句信息

管理视图 – 应用实践

- 获取缓冲池命中率

```
db2 "select substr(bp_name,1,30) as bp_name, data_hit_ratio_percent,
index_hit_ratio_percent, total_hit_ratio_percent from
sysibmadm.bp_hitratio where bp_name not like 'IBMSYSTEM%'“
```

- 获取Package Cache大小

```
db2 "with dbcfg1 as ( select int(value) as pckcachesz from
sysibmadm.dbcfg where name='pckcachesz') select pckcachesz as \"包缓
冲大小\", pkg_cache_lookups as  \"查询\", pkg_cache_inserts as \"插入\",
pkg_cache_num_overflows as \"溢出\", 100* pkg_cache_size_top /
(pckcachesz * 4096) as \"%包缓冲分配\" from
dbcfg1,sysibmadm.snapdb"
```

管理视图 – 应用实践

■获取执行成本最高的SQL

```
SELECT APPLICATION_HANDLE,ROWS_RETURNED,ROWS_READ  FROM SYSIBMADM.MON_CURRENT_SQL
```

■获取运行最长的SQL

```
SELECT SUBSTR(APPL_NAME,1,15) AS APPL_NAME, ELAPSED_TIME_MIN AS "Elapsed Min.", APPL_STATUS AS "Status ", SUBSTR(AUTHID,1,10) AS AUTH_ID , SUBSTR(INBOUND_COMM_ADDRESS,1,15) AS "IP Address“, SUBSTR(STMT_TEXT,1,30)  AS "SQL Statement"  FROM SYSIBMADM.LONG_RUNNING_SQL ORDER BY 2 DESC
```

■获取执行次数最多的SQL

```
SELECT NUM_EXECUTIONS AS "执行次数", AVERAGE_EXECUTION_TIME_S AS "平均时间秒", STMT_SORTS AS "排序次数", SORTS_PER_EXECUTION AS "每语句排序", SUBSTR(STMT_TEXT, 1, 30) AS "执行语句"  FROM SYSIBMADM.TOP_DYNAMIC_SQL  WHERE NUM_EXECUTIONS > 0 ORDER BY 2 DESC FETCH FIRST 5 ROWS ONLY
```

■获取排序次数最多的SQL

```
SELECT STMT_SORTS, SORTS_PER_EXECUTION, SUBSTR(STMT_TEXT, 1, 80) AS STMT_TEXT FROM SYSIBMADM.TOP_DYNAMIC_SQL ORDER BY STMT_SORTS DESC FETCH FIRST 5 ROWS ONLY
```

管理视图 – 应用实践

■获取LOCK WAIT等待时间

```
SELECT SUBSTR(AI.APPL_NAME, 1, 20) AS APPL_NAME,
SUBSTR(AI.PRIMARY_AUTH_ID, 1, 10) AS AUTH_ID,
 AP.LOCK_WAITS AS LOCK_WAITS, AP.LOCK_WAIT_TIME / 1000 AS "Total wait (s)",
 (AP.LOCK_WAIT_TIME / AP.LOCK_WAITS) AS "Avg Waits (ms)“
 FROM SYSIBMADM.SNAPAPPL_INFO AS AI, SYSIBMADM.SNAPAPPL AS AP
 WHERE AI.AGENT_ID = AP.AGENT_ID AND AP.LOCK_WAITS > 0
```

■获取LOCK CHAIN

```
SELECT SUBSTR(AI_H.APPL_NAME, 1, 10) AS "Hold App",
SUBSTR(AI_H.PRIMARY_AUTH_ID, 1, 10) AS "Holder",
SUBSTR(LW.APPL_NAME, 1, 10) AS "Wait App", SUBSTR(LW.AUTHID, 1, 10) AS "Waiter",
LW.LOCK_MODE, LW.LOCK_OBJECT_TYPE, SUBSTR(LW.TABNAME, 1, 10) AS "TabName",
SUBSTR(LW.TABSCHEMA, 1, 10) AS "Schema", TIMESTAMPDIFF(2,
CHAR(LW.SNAPSHOT_TIMESTAMP - LW.LOCK_WAIT_START_TIME)) AS "waiting (s)“
 FROM SYSIBMADM.LOCKWAITS AS LW, SYSIBMADM.SNAPAPPL_INFO AS AI_H
 WHERE LW.AGENT_ID_HOLDING_LK = AI_H.AGENT_ID
```

管理视图 – 应用实践

■获取锁升级、死锁和锁超时

```
SELECT SUBSTR(AI.APPL_NAME,1,10) AS APPLICATION,
SUBSTR(AI.PRIMARY_AUTH_ID,1,10) AS AUTHID,
INT (AP.LOCKS_HELD) AS "#Locks",INT(AP.LOCK_ESCALS) AS "Escalations",
INT(AP.LOCK_TIMEOUTS) AS "Lock Timeouts",
INT(AP.DEADLOCKS) AS "Deadlocks",
INT(AP.INT_DEADLOCK_ROLLBACKS) AS "Dlock Victim",
SUBSTR(INBOUND_COMM_ADDRESS,1,15) AS "IP Address"
FROM SYSIBMADM.SNAPAPPL AP,
SYSIBMADM.SNAPAPPL_INFO AI WHERE AP.AGENT_ID = AI.AGENT_ID
```

■监控全表扫描SQL

```
SELECT SUBSTR(AUTHID, 1, 10) AS AUTHID, SUBSTR(APPL_NAME, 1, 20) AS APPL_NAME,
PERCENT_ROWS_SELECTED FROM SYSIBMADM.APPL_PERFORMANCE
```

■获取数据库内存使用

```
SELECT POOL_ID,POOL_SECONDARY_ID, POOL_CUR_SIZE, POOL_WATERMARK FROM
SYSIBMADM.SNAPDB_MEMORY_POOL
```

提纲

- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top

事件监控器— 监控特定的事件

使用DB2事件监控来确定消耗资源最多的SQL

1. 创建事件监视器

```
db2 update mointor switches using statement on
db2 create event monitor sql_trace for statements write to table
db2 list tables for all | find /i “sql_trace”
```

```
STMT_SQL_TRACE              ADMINISTRATOR   T     2011-08-04-23.48.54.500002
```

2.激活监视器，并开始正常的数据库活动，直到想要的监控时段结束

```
db2 set event monitor sql_trace state=1
```

3.回到步骤1所打开的会话执行以下命令结束监控

```
db2 set event monitor sql_trace state=0
db2 terminate
```

事件监控—分析结果

■执行耗时最长的SQL语句

```
SELECT STMT_TEXT,(STOP_TIME-START_TIME) as “执行时间秒” FROM
STMT_SQL_TRACE WHERE STMT_OPERATION NOT IN (7,8,9,19) ORDER BY
DECIMAL(“执行时间秒”) DESC FETCH FIRST 10 ROWS ONLY
```

■执行次数最多的SQL

```
SELECT DISTINCT (STMT_TEXT), COUNT(1) AS “次数”  FROM STMT_SQL_TRACE
WHERE STMT_OPERATION NOT IN (7, 8, 9, 19)  GROUP BY STMT_TEXT  ORDER BY COUNT(1)
DESC  FETCH FIRST 10 ROWS ONLY
```

■最耗CPU时间的SQL语句

```
SELECT STMT_TEXT, USER_CPU_TIME AS "用户CPU秒" FROM STMT_SQL_TRACE  WHERE
STMT_OPERATION NOT IN (7, 8, 9, 19) ORDER BY "用户CPU秒" DESC FETCH FIRST 10 ROWS
ONLY
```

■总排序时间最长的SQL语句

```
SELECT STMT_TEXT,TOTAL_SORT_TIME “排序时间秒” FROM STMT_SQL_TRACE WHERE
STMT_OPERATION NOT IN (7,8,9,19) ORDER BY DECIMAL(TOTAL_SORT_TIME) DESC FETCH
FIRST 10 ROWS ONLY
```

提纲

- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top

强大的db2pd

- -osinfo → 操作系统信息
- -dbmcfg → DBM配置参数
- -dbcfg → DB配置参数
- -edus → 监视edu
- -mempools -memsets → 内存池和内存集
- -utilities → 工具 (backup, runstats, reorg, restore, load)
- -applications → 应用程序
- -transactions →事务
- -bufferpool → 缓冲池
- -logs → 日志
- -locks → 锁
- -tablespaces → 表空间和容器
- -dynamic → 动态SQL
- -dbptnmem → 实例所消耗的内存总量
- -tcbstats → 索引利用率

强大的db2pd:系统信息

db2pd -osinfo

```
Operating System Information:

OSName:   Linux
NodeName: main.zhutr.gicp.net
Version:  2
Release:  6
Machine:  x86_64

CPU Information:
TotalCPU    OnlineCPU   ConfigCPU   Speed(MHz)  HMTDegree  Cores/Socket
4           4           4           2400        1           4

Physical Memory and Swap (Megabytes):
TotalMem    FreeMem     AvailMem    TotalSwap   FreeSwap
8002        137         n/a         3907        3890

Virtual Memory (Megabytes):
Total       Reserved    Available   Free
11909       n/a         n/a         4027

Message Queue Information:
MsgSeg      MsgMax      MsgMap      MsgMni      MsgTql      MsgMnb      MsgSsz
n/a         65536       65536       15994       65536       65536       16

Shared Memory Information:
ShmMax              ShmMin              ShmIds      ShmSeg
8389881856          1                   4096        4096

Semaphore Information:
SemMap      SemMni      SemMns      SemMnu      SemMsl      SemOpm      SemUme      SemUsz      SemVmx      SemAem
256000      2048        256000      256000      250         32          n/a         20          32767       32767
```

强大的db2pd -bufferpool

关键指标:命中率(HitRatio)

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 1 days 00:42:44

Bufferpools:
First Active Pool ID      1
Max Bufferpool ID         1
Max Bufferpool ID on Disk 1
Num Bufferpools           5

Address    Id  Name          PageSz   PA-NumPgs  BA-NumPgs  BlkSize   NumTbsp   PgsToRemov CurrentSz  PostAlter  SuspndTSCt
0x7F91F570 1   IBMDEFAULTBP  8192     250        0          0         7         0          250         250        0
0x7F91F290 4099 IBMSYSTEMBP32K 32768  16         0          0         0         0          16          16         0

Bufferpool Statistics for all bufferpools (when BUFFERPOOL monitor switch is ON)
:

BPID DatLRds  DatPRds  HitRatio TmpDatLRds TmpDatPRds HitRatio IdxLRds  IdxPRds  HitRatio TmpIdxLRds TmpIdxPRds HitRatio
1    79635    444      99.44%   7          0          100.00%   38699    564      98.54%   0          0          00.00%
4099 0        0        00.00%   0          0          00.00%    0        0        00.00%   0          0          00.00%

BPID DataWrts  IdxWrts  DirRds   DirRdReqs  DirRdTime  DirWrts  DirWrtReqs DirWrtTime
1    0         0        29480    2018       60385      1068     5          33537
4099 0         0        0        0          0          0        0          0

BPID AsDatRds  AsDatRdReq AsIdxRds  AsIdxRdReq AsRdTime  AsDatWrts  AsIdxWrts  AsWrtTime
1    21        12         5         4          525       0          0          0
4099 0         0          0         0          0         0          0          0

BPID TotRdTime  TotWrtTime VectIORds  VectIOReq  BlockIORds BlockIOReq FilesClose NoVictAvl  UnRdPFetch
1    8479       0          0         0          0         0          0          505
4099 0          0          0         0          0         0          0          0
```

db2pd示例:每天下午系统神秘的变慢

- 用户报告每天下午1点系统整体性能下降,并声明没有改动任何东西
- 运行 runstats并不解决问题，排除统计信息过时这种情况
- 监控和检查各种数据库性能指标比如缓冲池命中率，锁等待，等等，没有发现性能瓶颈
- “vmstat” 显示high user cpu，这是性能分析的第一线索:

```
procs -----------memory---------- ---swap-- -----io---- -system--      -----cpu------
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs                    us sy id wa st
 2  0 520800  60916   3800 438032   32  132   393   183  280  451 53  8 33  6  0
 1  0 520800  60636   3808 438032    0    0     0    48  259  376          57  4 37  2  0
 3  0 520800  60652   3824 438032    0    0     0     8  263  393   54  8 34  4  0
 1  0 520800  60776   3832 438032    0    0     0    26  258  371          60  1 39  0  0
```

db2pd示例:每天下午系统神秘的变慢

- 用户CPU高表明这是由应用程序空间(DB2)的活动导致的，而不是在操作系统中
- 用 “db2pd –edus” 查看每个线程的cpu 使用率:

EDU ID	TID	Kernel TID	EDU Name	USR (s)	SYS (s)
.....					
21	2946493344	10227	db2agent (DS2)	2.200000	0.900000
20	2947541920	10226	db2lfr (DS2)	0.000000	0.000000
19	2948590496	10225	db2loggw (DS2)	1.700000	0.600000
18	2949639072	10224	db2loggr (DS2)	2.170000	1.680000
17	2950687648	10214	db2agent (DS2)	29.310000	4.600000
16	2951736224	10201	db2resync	0.000000	0.000000
15	2952784800	10200	db2tcpcm	0.310000	0.450000

EDU id 17 使用了大量的用户CPU，把它找出来！

db2pd示例:每天下午系统神秘的变慢

- 使用“db2pd –db ds2 –applications” 找到EDU ID 17服务的那个应用程序:

Address	App-Handl	[nod-index]	Num-Agents	Coor-EDUID	Status	L-Anch-ID	L-Stmt-UID	Appid
0x20A55FD0	12	[000-00012]	1	30	ConnectCompleted	0	0	*LOCAL.DB2.091015124129
0x20A50060	11	[000-00011]	1	29	ConnectCompleted	0	0	*LOCAL.DB2.091015124128
0x20A45FD0	10	[000-00010]	1	28	ConnectCompleted	0	0	*LOCAL.DB2.091015124127
0x20A40060	9	[000-00009]	1	27	ConnectCompleted	0	0	*LOCAL.DB2.091015124126
0x20A06C00	8	[000-00008]	1	26	ConnectCompleted	0	0	*LOCAL.DB2.091015124125
0x20A00060	7	[000-00007]	3	17	PerformingLoad	937	1	*LOCAL.johnh.091015130001

- 应用程序handle 7 正在由EDU id 17服务.
- Status域显示的LOAD操作解释了这个问题。
- Load操作是很耗资源的操作，导致系统的整体性能下降。
- 无经验用户johnh在每天下午调度了LOAD操作。

提纲

- 性能监控方法学
- 操作系统监控
- 快照监控与性能分析
- 管理视图应用实践
- 事件监控器
- db2pd应用和案例
- db2mtrk & db2top

db2mtrk 监控内存

```
C:\Documents and Settings\Administrator>db2mtrk -i -d -v
在 2011/08/06 16:28:49 跟踪内存

用于实例的内存

   Other Memory 的大小为 13238272 字节
   FCMBP Heap 的大小为 786432 字节
   Database Monitor Heap 的大小为 327680 字节
   总计: 14352384 字节

用于数据库 SAMPLE   的内存

   Backup/Restore/Util Heap 的大小为 65536 字节
   Package Cache 的大小为 1245184 字节
   Other Memory 的大小为 131072 字节
   Catalog Cache Heap 的大小为 327680 字节
   Buffer Pool Heap (1) 的大小为 2424832 字节
   Buffer Pool Heap (System 32k buffer pool) 的大小为 851968 字节
   Buffer Pool Heap (System 16k buffer pool) 的大小为 589824 字节
   Buffer Pool Heap (System 8k buffer pool) 的大小为 458752 字节
   Buffer Pool Heap (System 4k buffer pool) 的大小为 393216 字节
   Shared Sort Heap 的大小为 65536 字节
   Lock Manager Heap 的大小为 17367040 字节
   Database Heap 的大小为 15007744 字节
   Application Heap (393) 的大小为 65536 字节
   Application Heap (392) 的大小为 65536 字节
   Application Heap (391) 的大小为 65536 字节
   Application Heap (390) 的大小为 131072 字节
   Application Heap (389) 的大小为 65536 字节
   Application Heap (388) 的大小为 65536 字节
   Application Heap (387) 的大小为 65536 字节
   Application Heap (386) 的大小为 65536 字节
   Applications Shared Heap 的大小为 720896 字节
   总计: 40239104 字节
```

–db2mtrk –i　显示当前实例内存使用情况

–db2mtrk –i –v　显示当前实例的内存的详细信息

–db2mtrk –d　显示当前数据库的内存使用情况

–db2mtrk –d –v　显示数据库的内存使用情况的详细信息

–db2mtrk –p　显示代理进程专用内存使用率

–db2mtrk –h　显示帮助信息

-m 参数用于显示最大的内存使用上限

-w 参数选项用于显示使用过程中内存达到的最大值，即watermark

-r 参数选项用于重复显示，其中
interval指定以秒为单位的重复间隔
count指定间隔的次数

-v 详细输出

也能用db2pd –memsets –mempools做同样的事！

db2top – linux平台常用的工具

db2top -d sample

```
[\]20:02:16,refresh=2secs(0.002)
Linux,part=[1/1],DB2INST1:SAMPLE
[d=Y,a=N,e=N,p=ALL]
[qp=off]
######  ######   #####  ####### ####### ######   For help type h or
#     # #     # #     #    #    #     # #     #  db2top -h: usage
#     # #     #       #    #    #     # #     #
#     # ######   #####     #    #     # ######   Status: Active
#     # #     # #          #    #     # #        Uptime: 03h:10m:32s
#     # #     # #          #    #     # #        Last backup
######  ######  #######    #    ####### #        None
DB2 Interactive Snapshot Monitor V2.0
Use these keys to navigate:
d - Database            l - Sessions            a - Agent
t - Tablespaces         b - Bufferpools         T - Tables
D - Dynamic SQL         U - Locks               m - Memory
s - Statements          p - Partitions          u - Utilities
A - HADR                F - Federation          B - Bottlenecks
J - Skew monitor        q - Quit
```

文本交互界面，能一目了然监控everything

db2top – 动态SQL

■输入D查看动态SQL

```
[-]21:13:06,refresh=2secs(0.001)                SQL                Linux,part=[1/1],DB2INST1:SAMPLE
[d=Y,a=N,e=N,p=ALL]                                                                          [qp=off]

  SQL_Statement              Sql                                  Num       Exec        Avg        Cpu
  HashValue                  Statement (30 first char.)       Execution       Time    ExecTime       Time
  -------------------------- -------------------------------- --------- ---------- ---------- ----------
  00000001710740561843197885 SELECT evmon.xmlreport FROM TA          45   0.764746   0.016994   0.676125
  00000009419268137772287756 select case(upper(reg_var_valu          45   0.174392   0.003875   0.030654
  00000000011654372868272295 INSERT INTO pedemo.res_b VALUE           1   0.001267   0.001267   0.001243
  00000000086255143002562133 UPDATE pedemo.res_b SET descri           4   0.005657   0.001414   0.005529
  00000001007369034269157 61 INSERT INTO pedemo.res_a VALUE           1   0.001232   0.001232   0.001213
  00000001027492070293278 71 UPDATE pedemo.res_b SET descri           4   0.004186   0.001046   0.004085
  00000001302133528931264 62 set current schema OPM                  22   0.003326   0.000151   0.002799
  00000001775236563852735 22 UPDATE pedemo.res_b SET descri           2   0.002602   0.001301   0.002556
  00000001780107693128077 45 UPDATE pedemo.res_b SET descri           2   0.002819   0.001409   0.002727
  00000001857692464615474 76 UPDATE pedemo.res_b SET descri           2   0.002798   0.001399   0.002734
  00000001874345172100167 29 UPDATE pedemo.res_b SET descri           2   0.002798   0.001399   0.002744
  00000002087778178427370 90 UPDATE SYSTOOLS.HMON_ATM_INFO            1   0.002753   0.002753   0.002716
  00000002713125443669974 15 UPDATE pedemo.res_b SET descri           2   0.002636   0.001318   0.002599
  00000002727559305677045 05 UPDATE pedemo.res_b SET descri           2   0.002775   0.001387   0.002735
  00000003527926417598723 08 SELECT COUNT( * ) FROM SYSTOOL           1   0.001760   0.001760   0.001733
  00000006201643433062934 80 UPDATE pedemo.res_b SET descri           2   0.002651   0.001325   0.002600
  00000006208682907137940 92 UPDATE pedemo.res_b SET descri           2   0.002769   0.001384   0.002719
Quit: q, Help: h             Dynamic SQL 1011 (Cached=1011), L: Query Text                    db2top 2.0
```

```
db2top -d sample  -b  D  -o top.sql -m 1
```

■以后台方式查看动态SQL，输出到文件

db2top还支持输出到文件

小结

- 操作系统层面的监控便于我们发现性能瓶颈所在，作为监控调优的起点。
- DB2监控输出包含很多有用的数据，但是其中的某些更便于我们定位问题
  - 缓冲池命中率，读数据行的效率(rows read/selected), 排序的溢出比率等
  - 优先考虑这些重要的指标将对性能调优起到事半功倍的效果
- 管理视图提供了一个很好的存储和分析历史监控数据的SQL界面。
- db2pd 是一个强大、方便的工具，可窥视DB2内部运行情况。
- 事件监控器中的DB2活动和语句监控提供了综合行的性能数据。
- 在实践过程中，各种工具可以配合使用，比如以快照或db2pd为基础，配置其他便捷工具，以达到最好的性能监控效果。

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间