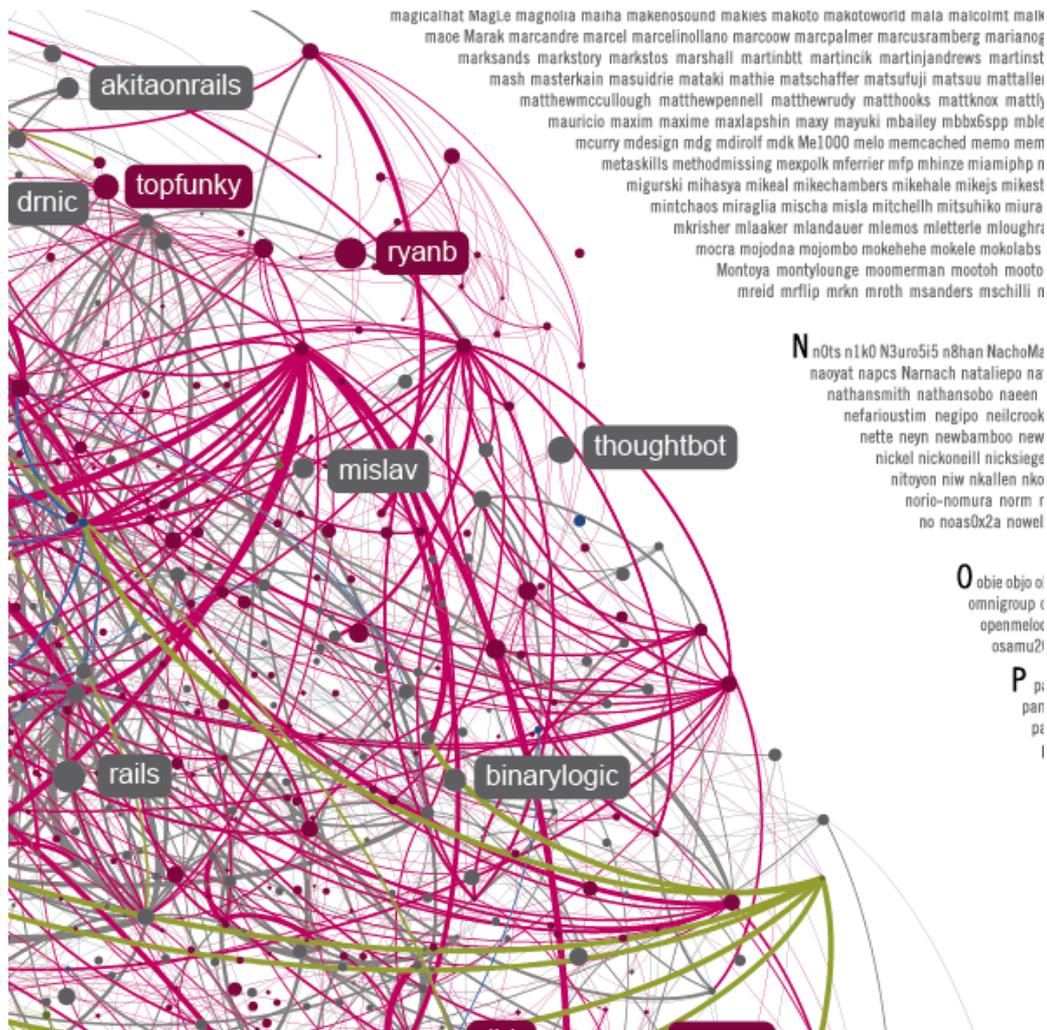


# DB2设计与性能优化

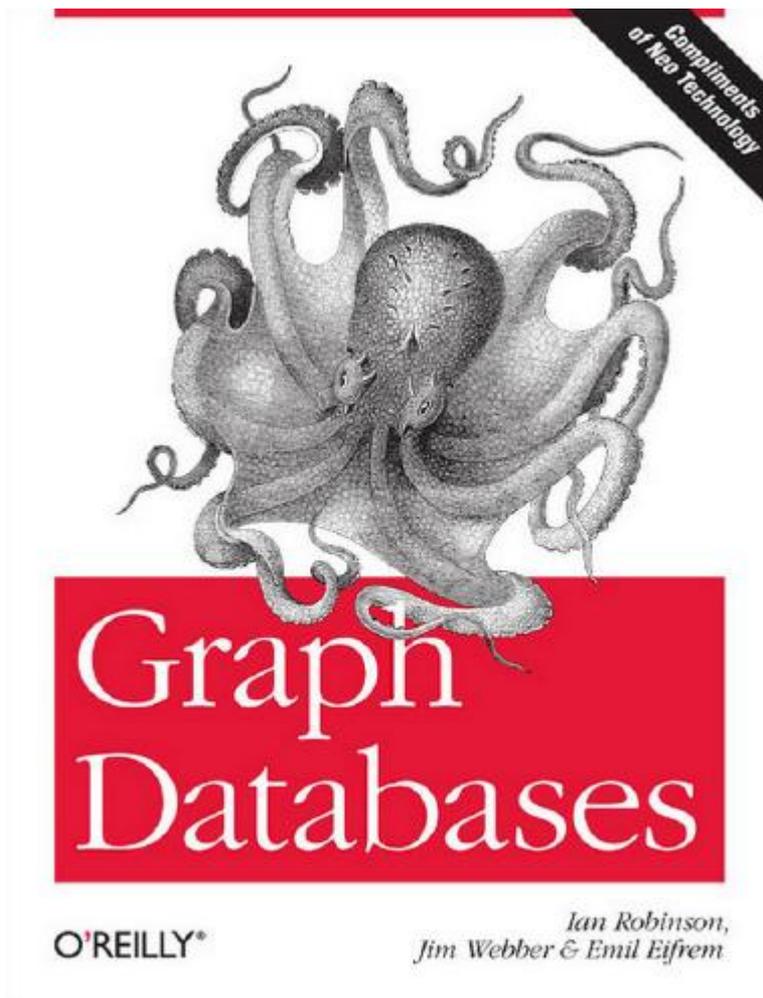
## 第4周



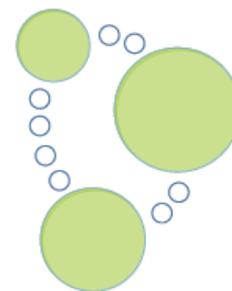
**【声明】** 本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

<http://edu.dataguru.cn>



## The Neo4j Manual



- 操作系统参数
- 注册表变量
- 数据库配置
- 初始化配置
- **STMM**自动内存管理

- 
- 操作系统参数 ( 操作系统内核参数 )
  - 注册表变量
  - 数据库配置
  - 初始化配置
  - **STMM**自动内存管理

- VMO – 虚拟内存管理

- lru\_file\_repage: 设置为0
- maxperm: 设置为20

```
vmo -p -o lru_file_repage=0 -o  
maxclient%=20  
vmo -p -o maxperm%=20 -o minperm%=3
```

- AIO – 异步I/O

- maxservers: 根据系统压力适当调大
- maxreqs: MAX( NUM\_IIOCLEANERSx256, 4096 )

```
chdev -l aio0 -a  
maxservers=100  
chdev -l aio0 -a  
maxreqs=4096
```

- 开启IOCP

- 启用IOCP后, 可用来处理 AIX 操作系统上的异步 I/O (AIO) 请求, 从而改进整体系统性能

```
# smitty iocp
```

- 内核参数/etc/sysctl.conf

```
#Example for a computer with 16GB of RAM:  
kernel.shmmni=4096  
kernel.shmmax=17179869184  
kernel.shmall=8388608  
#kernel.sem=<SEMMSL> <SEMMNS> <SEMOPM>  
<SEMMNI>  
kernel.sem=250 256000 32 4096  
kernel.msgmni=16384  
kernel.msgmax=65536  
kernel.msgmnb=65536
```

# Solaris/HP-UX内核参数

- 针对Solaris/HP-UX的内核参数配置工具
  - db2osconf
- Solaris内核参数配置文件
  - /etc/system
- HP-UX内核参数配置方法
  - HP-UX sam工具

```
# /opt/IBM/db2/V9.7/bin/db2osconf

set msgsys:msginfo_msgmni = 7168
set semsys:seminfo_semmni = 8192
set shmsys:shminfo_shmmax = 123655598899
set shmsys:shminfo_shmmni = 8192

Total kernel space for IPC:
1.12MB (shm) + 3.00MB (sem) + 2.48MB (msg) == 6.60MB (total)
```

- 操作系统参数
- ➔ ■ 注册表变量
- 数据库配置
- 初始化配置
- **STMM**自动内存管理

# 注册表变量操作

- 查看已设置注册表变量
  - db2set -all
- 查看所有注册表变量
  - db2set -lr
- 修改注册表变量
  - db2set variable=value
- 注销注册表变量
  - db2set variable=

## 注册表变量(DB2\_PARALLEL\_IO)

- DB2\_PARALLEL\_IO参数定义容器的并行度
  - 容器的并行度定义了每个容器对应几个物理磁盘
  - 当prefetch定义为automatic时生效
- db2set DB2\_PARALLEL\_IO=\*:5
  - 所有表空间容器的并行度为5
- db2set DB2\_PARALLEL\_IO=\*:5,2:8
  - 表空间2的容器的并行度为8
  - 其他表空间容器的并行度为5

## 注册变量(并发相关)

- 下列注册表变量与并发性相关
  - DB2\_EVALUNCOMMITTED
    - 正常模式下，DB2先加锁，再检查数据
    - 设置该参数后，先检查数据，再加锁
  - DB2\_SKIPINSERTED
    - 在CS与RS隔离级别下，将忽略新插入的未提交数据
  - DB2\_SKIPDELETED
    - 在CS与RS隔离级别下，将忽略删除的未提交数据
- 设置上述参数，需对应用程序进行测试

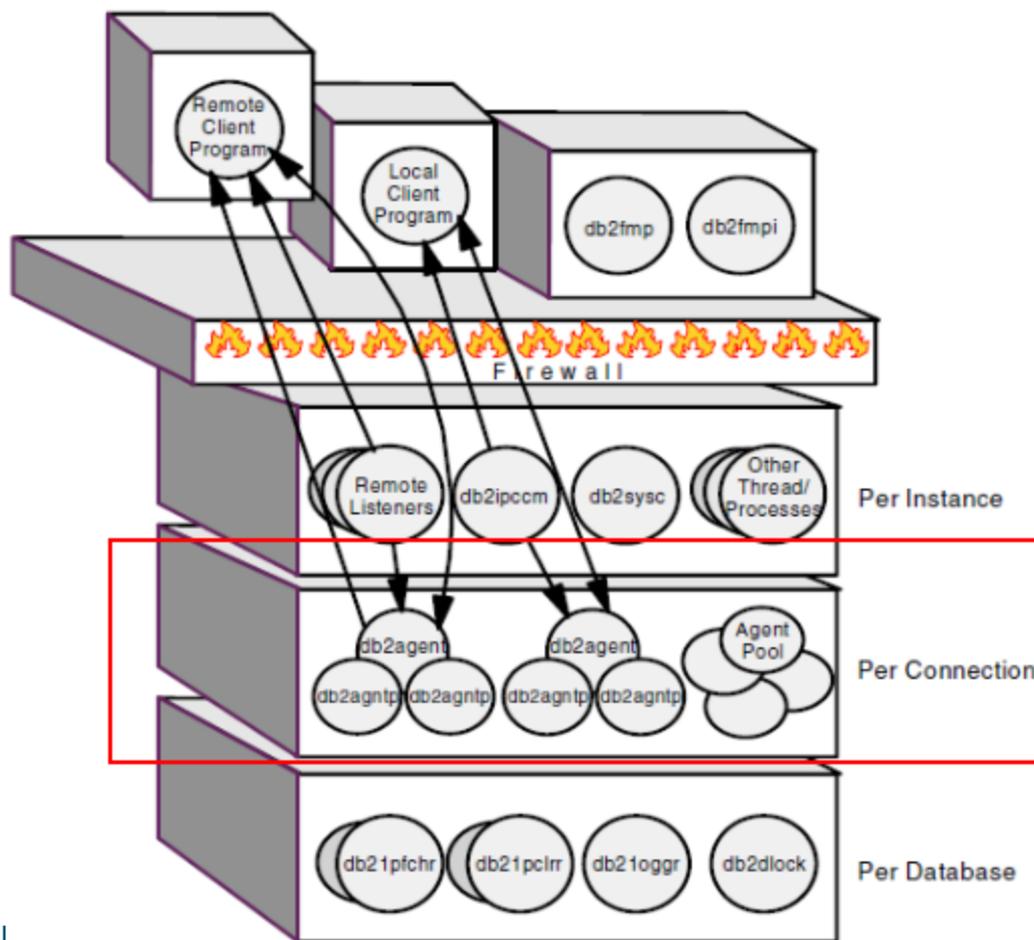
- 操作系统参数
- 注册表变量
- ➔ ■ 数据库配置
  - 初始化配置
- **STMM**自动内存管理

## 数据库配置参数的内容

- **DB2代理**
- **缓冲池**
- **DBHEAP**
- **Package Cache**
- **Catalog Cache**
- **Utility Heap**
- **Application Heap**
- **Statement Heap**
- **Statistics Heap**
- **SORTHEAP**
- **日志**
- **锁**
- **优化器**
- **异步写**
- **预取**

# DB2代理(agent)

- 代理接收并执行client发送的SQL语句
- 未启用连接集中器
  - 一个连接对应一个代理
- 启用连接集中器
  - 多个连接共用一个代理



# 代理相关参数

- DBM CFG
  - MAX\_COORDAGENTS
  - MAX\_CONNECTIONS
  - NUM\_POOLAGENTS
  - NUM\_INITAGENTS
- DB CFG
  - MAXAPPLS

- **MAX\_COORDAGENTS**
  - 最大协调代理进程数
- **NUM\_POOLAGENTS**
  - 代理池中空闲代理的数量
- **MAX\_CONNECTIONS**
  - 最大客户机连接数
- **NUM\_INITAGENTS**
  - 代理池中的初始代理数量
- **MAXAPPLS**
  - 一个实例只有一个数据库，建议与 **MAX\_CONNECTIONS**保持一致
  - 多个数据库：**MAXAPPLS**之和不能超过**MAX\_CONNECTIONS**

- 启用连接集中器  
(`MAX_CONNECTIONS > MAX_COORDAGENTS`)
  - 优点：对于频繁创建连接的应用，可以提升性能
  - 缺点：一个代理要为多个连接服务，有时候会出现代理资源不足
- 不启用连接集中器  
(`MAX_CONNECTIONS = MAX_COORDAGENTS`)
  - 优点：对于具有连接池功能的应用程序，可以提升性能
  - 缺点：过多的连接会消耗大量的进程或线程资源

## 代理参数监控

- **db2 get snapshot for db on <dbname>**
  - **Maximum agents associated with applications** 该值接近参数 **MAX\_COORDAGENTS**，说明高峰期代理资源紧张
  - **Application connects** 该值过高说明应用程序连接池配置不当，导致频繁创建连接
- 应用程序日志
  - 如果出现 **SQL1040N** 说明代理资源紧张
- 设置 **max\_coordagents** 的意义
  - 防止连接过多，导致系统资源耗尽

- BUFFERPOOL的大小
  - 使用alter bufferpool命令调整大小
- 查看BUFFERPOOL的大小
  - `select substr(bpname,1,20), npages from syscat.bufferpools`
- BUFFERPOOL大小的调整原则
  - OLTP系统保证命中率尽量接近100%
  - BUFFERPOOL比表空间还大没有意义
  - 不要调整的过大，使操作系统尽量少的使用SWAP空间为宜

# 缓冲池的性能指标-命中率

- BUFFERPOOL命中率的计算公式
  - $(\text{逻辑读} - \text{物理读} / \text{逻辑读}) * 100\%$
- BUFFERPOOL命中率的计算方法
  - 通过SYSIBMADM.BP\_HITRATIO视图直接查看
  - 通过SYSIBMADM.SNAPBP视图或SNAPSHOT for BUFFERPOOL命令计算

# 缓冲池命中率的计算(1)

## ■ BP\_HITRATIO视图

```
SELECT
  substr(DB_NAME,1,8) as DB_NAME,
  substr(BP_NAME,1,20) as BP_NAME,
  TOTAL_HIT_RATIO_PERCENT as TOTAL_RATIO,
  DATA_HIT_RATIO_PERCENT as DATA_RATIO,
  INDEX_HIT_RATIO_PERCENT as INDEX_RATIO,
  XDA_HIT_RATIO_PERCENT as XDA_RATIO
FROM SYSIBMADM.BP_HITRATIO
```



DB_NAME	BP_NAME	TOTAL_RATIO	DATA_RATIO	INDEX_RATIO	XDA_RATIO
TEST	IBMDEFAULTBP	64.00	66.57	61.89	-
TEST	IBMSYSTEMBP4K	-	-	-	-
TEST	IBMSYSTEMBP8K	-	-	-	-
TEST	IBMSYSTEMBP16K	-	-	-	-
TEST	IBMSYSTEMBP32K	-	-	-	-

## ■ SNAPSHOT命令或SNAPBP视图

```
[db2inst2@beijing ~]$ db2 get snapshot for bufferpools on test
```

### Bufferpool Snapshot

```
Bufferpool name           = IBMDEFAULTBP
Database name             = TEST
Database path             =
/home/db2inst2/db2fs/db2inst2/NODE0000/SQL00001/
Input database alias     = TEST
Snapshot timestamp       = 01/01/2012
09:41:32.411385

Buffer pool data logical reads   = 795
Buffer pool data physical reads  = 243
Buffer pool index logical reads  = 912
Buffer pool index physical reads = 338
```



```
DATA_RATIO=(795-243)/795=69.43%
INDEX_RATIO=(912-338)/912=62.94%
```

- 每个数据库会分配一个DBHEAP
- DBHEAP中包含如下信息
  - 对象控制信息(tables, indexes, table spaces, buffer pools)
  - 日志缓冲区(LOGBUFSZ)
  - Utilities的临时内存
- DBHEAP大小的估算
  - 每个tablespace占10K
  - 每个table占4K

- DBHEAP相关的性能指标
  - DBHEAP Current size 当前使用的大小
  - DBHEAP High water mark 历史最大值(高水位)
  - DBHEAP Configured size 配置的大小
- 应确保POOL\_WATERMARK不超过POOL\_CONFIG\_SIZE

```
SELECT
  substr(DB_NAME, 1, 10) as DB_NAME,
  POOL_CUR_SIZE,
  POOL_WATERMARK,
  POOL_CONFIG_SIZE
FROM
  SYSIBMADM.SNAPDB_MEMORY_POOL
WHERE
  POOL_ID='DATABASE'
```



DB_NAME	POOL_CUR_SIZE	POOL_WATERMARK	POOL_CONFIG_SIZE
SAMPLE	19791872	19791872	31653888

1 record(s) selected.

# PCKCACHESZ (Package Cache)

## ■ 什么是Package

- Package是SQL/XQuery编译后的可执行方案
- 静态SQL是预编译的，其方案存储在系统表中
- 动态SQL在执行的时候才进行编译
- 请不要与Oracle中的Package混淆，Oracle中的Package对应DB2中的Module

## ■ Package Cache缓存

- 缓存动态与静态SQL/XQuery的执行方案
- 相同SQL被执行多次的时候，可以提升性能
- 大小由PCKCACHESZ参数决定

# Package Cache的性能指标-命中率

- Package Cache命中率的计算

```
select
  1-(float(PKG_CACHE_INSERTS)/PKG_CACHE_LOOKUPS) as
  RATIO,
  PKG_CACHE_NUM_OVERFLOWS
from SYSIBMADM.SNAPDB
```



```
RATIO                PKG_CACHE_NUM_OVERFLOWS
-----
+6.800000000000000E-001      0

1 record(s) selected.
```

- Package Cache其他参考性能指标

- **PKG\_CACHE\_NUM\_OVERFLOWS** 溢出次数：当Package Cache不足以存放当前正在执行的SQL方案时，会将内存中的内容溢出到磁盘

- 确保命中率接近100%，并且尽量避免溢出

# CATALOGCACHE\_SZ (Catalog Cache)

- 每个数据库会分配一个Catalog Cache
- Catalog Cache用来缓存系统编目信息
- Catalog Cache能够提升下列操作的性能
  - 绑定package、编译SQL/XQuery
  - 数据库权限检查
- DB CFG参数CATALOGCACHE\_SZ定义了内存上限
- 确保命中率接近100%，并且尽量避免溢出

```
select
    1-(float(CAT_CACHE_INSERTS)/CAT_CACHE_LOOKUPS) as RATIO,
    CAT_CACHE_OVERFLOWS
from SYSIBMADM.SNAPDB
```



RATIO	CAT_CACHE_OVERFLOWS
+6.500000000000000E-001	0

1 record(s) selected.

# UTIL\_HEAP\_SZ (Utility Heap)

- UTIL\_HEAP\_SZ参数控制Utility Heap内存上限
- 下列参数会使用Utility Heap
  - backup
  - restore
  - Load
- Utility Heap在工具运行结束后，内存就会回收，不会常驻内存
- Utility Heap相关的性能指标
  - Utility Heap Current size 当前使用的大小
  - Utility Heap High water mark 历史最大值(高水位)
  - Utility Heap Configured size 配置的大小

# Utility Heap性能指标的监控(1)

- SNAPDB\_MEMORY\_POOL视图

```
SELECT
    substr(DB_NAME, 1, 10) as DB_NAME,
    POOL_CUR_SIZE,
    POOL_WATERMARK,
    POOL_CONFIG_SIZE
FROM
    SYSIBMADM.SNAPDB_MEMORY_POOL
WHERE
    POOL_ID='UTILITY'
```



DB_NAME	POOL_CUR_SIZE	POOL_WATERMARK	POOL_CONFIG_SIZE
SAMPLE	65536	65536	20512768

1 record(s) selected.

# applheapsz (Application Heap)

- 每个连接拥有自己的Application Heap，存放连接相关的(比如当前正在执行的SQL等)信息

```

$ db2 get snapshot for db on sample
... ..
Node number                = 0
  Memory Pool Type         = Application Heap
    Secondary ID           = 320
      Current size (bytes) = 65536
        High water mark (bytes) = 65536
          Configured size (bytes) = 1048576
  
```

- 性能指标
  - 确保多数应用程序的Application Heap监控结果中，High water mark 小于Configured size
  - 当应用程序接收到SQL0954C错误时，说明Application Heap内存不足

# Stmtheap(Statement Heap)

- SQL/XQuery编译时使用的工作空间
- 当接收到SQL0101N The statement is too long or too complex.错误时，需要增加该参数

# STAT\_HEAP\_SZ (Statistics Heap)

- Statistics Heap用来在执行runstats时存放临时信息
- STAT\_HEAP\_SZ参数控制该heap的内存上限
- 下列因素决定该内存的需求量
  - runstats时指定Detailed选项，将需要更多内存
  - 收集信息的列越多，需要的内存越多
- 该内存在执行runstats时分配，runstats运行完成后释放

- 当出现下列错误或警告时，说明Statistics Heap所需的内存不足，需要增加STAT\_HEAP\_SZ参数的大小
  - SQL2312N Statistics heap size is too small to carry out the operation. Suggested heap size is *num* pages.
  - SQL2313W All available memory in the statistics heap has been used. Statistics were collected on the first *n1* rows out of a total of *n2* rows.

# SORTHEAP排序堆大小

- SORTHEAP排序堆大小
  - SORTHEAP为代理私有内存，在排序时分配，排序完成后就会释放
  - 当排序数据量较大而SORTHEAP不够用时，将产生排序溢出，部分数据将会被写到临时表空间中
- OLTP系统
  - 由于并发量较大，SORTHEAP需要分配合适
- OLAP系统
  - 根据物理内存数量与并发数，可将SORTHEAP调整大一些

# 对排序内存加以限制

- 高并发系统，排序将消耗大量内存，不加以限制，可能导致操作系统物理内存耗尽而宕机
- **SORTHEAP**可以限制一个连接的排序内存，而对于总排序内存，需要使用以下参数加以限制
  - **SHEAPTHRES** 实例范围内的私有排序总内存限制
  - **SHEAPTHRES\_SHR**数据库范围内的共享排序总内存限制
  - 当**SHEAPTHRES**设置为>0时，将使用**SHEAPTHRES**限制排序总内存；当**SHEAPTHRES**设置为=0时，将使用**SHEAPTHRES\_SHR**限制排序总内存
  - 建议使用共享排序

# 排序的性能指标-排序溢出率

- 排序发生溢出时，系统性能将下降
- 通过SNAPSHOT for DB命令监控排序溢出率
  - 排序溢出率=Sort overflows / Total sorts

```
$ db2 get snapshot for db on sample|grep -i sort
Total Private Sort heap allocated      = 0
Total Shared Sort heap allocated      = 0
Shared Sort heap high water mark     = 29
Post threshold sorts (shared memory)  = 0
Total sorts                           = 2
Total sort time (ms)                  = 22
Sort overflows                         = 0
Active sorts                          = 0
```

- 排序溢出率过高，可以适当调整SORTHEAP, SHEAPTHRES, SHEAPTHRES\_SHR等参数

# 排序问题优化思路

- 调整排序参数，分配适当的内存给排序堆
- 排序问题的根源在于索引与SQL
  1. 创建适当的索引，尽量减少排序
  2. 优化SQL语句，使用合理的排序
  3. 分析业务，消除不必要的排序

# 日志性能相关的参数

- 日志缓冲区
  - LOGBUFSZ 建议配置为合适的大小
- 日志数量与大小
  - LOGFILSIZ 日志文件大小
  - LOGPRIMARY 主日志文件个数
  - LOGSECOND 辅日志文件个数
  - 确保  $\text{LOGFILSIZ} \times (\text{LOGPRIMARY} + \text{LOGSECOND})$  满足所有事务所需要的日志总和
- MINCOMMIT 组落实
  - 控制多个应用程序一次提交，只适用于事务小而频繁的系统
  - 这个参数已经不推荐使用

# 日志性能监控

- 日志使用率高水位

```
$ db2 get snapshot for db on sample|grep -i "Log space"
Log space available to the database (Bytes)= 17334246
Log space used by the database (Bytes)      = 3045754
Maximum secondary log space used (Bytes)   = 1886260
Maximum total log space used (Bytes)       = 14114260
```

- 日志使用率高水位=  $14114260 / (17334246 + 3045754) = 69.26\%$
- 日志使用率高水位接近100%，说明系统极有可能发生“日志满”错误，需要增加日志总大小

- 日志缓冲区满

```
$ db2 get snapshot for db on sample|grep -i "Number log buffer full"
Number log buffer full                = 0
```

- 经常发生日志缓冲区满，需要增加LOGBUFSZ

# 锁相关参数

- **LOCKLIST**锁列表的内存大小
  - 数据库内的所有锁信息都存放在该内存中
- **MAXLOCKS**每个连接能使用的锁列表的最大比例
  - 限制应用程序对锁内存的使用
- **LOCKTIMEOUT**锁超时时间
  - 当所等待达到这一时间后，将会报错，事务将会滚

- 锁升级
  - **LOCKLIST**达到上限后，多个行级锁转换为一个表级锁，以节省内存
- 锁超时
  - 所等待时间达到**LOCKTIMEOUT**后，应用程序失败而回滚
- 死锁
  - 相互等待对方释放锁资源

# 锁相关的性能指标

- 根据应用程序属性，判断下列指标是否在可接受范围
  - 每小时死锁数量
  - 每小时锁超时数量
  - 每小时锁升级数量

```
$ db2 get snapshot for db on sample|grep -i lock
Locks held currently           = 0
Lock waits                     = 0
Time database waited on locks (ms) = 0
Lock list memory in use (Bytes) = 19200
Deadlocks detected             = 0
Lock escalations               = 2
Exclusive lock escalations     = 2
Agents currently waiting on locks = 0
Lock Timeouts                  = 0
```

- **INTRA\_PARALLEL** 启用分区内并行性
  - 一个SQL语句将由多个代理并行执行
  - 在OLTP环境，建议不打开该参数
  - 在OLAP环境，建议打开该参数
- **DFT\_QUERYOPT** 缺省查询优化类
  - 指导优化器对SQL语句的优化程度
  - 取值范围0-9，默认5
  - 数值越大，优化程度越高，编译时间越长
  - **CURRENT QUERY OPTIMIZATION**语句或bind命令的参数可以覆盖该缺省值

- **NUM\_IOCLEANERS**参数定义了异步页清除程序数目
- 异步页清除程序的功能
  - 将bufferpool中更新(包括insert/update/delete)的“脏数据”写入磁盘
- **NUM\_IOCLEANERS**参数的建议配置
  - 比物理磁盘数多一点
  - 该参数不会占用过多资源，因此可以配置稍大一些

# 异步写-性能指标

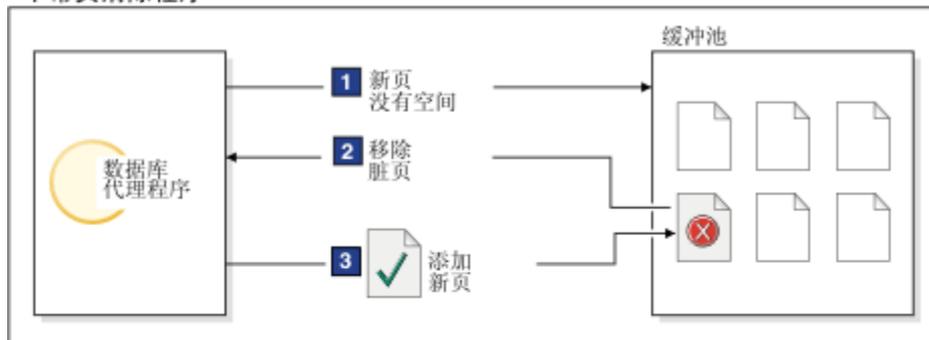
- 通过SNAPSHOT for DB命令监控下列元素
  - Buffer pool data writes 数据写的次数
  - Asynchronous pool data page writes 数据异步写的次数
  - Buffer pool index writes 索引写的次数
  - Asynchronous pool index page writes 索引异步写的次数
- 当满足下面的条件时，说明数据异步写没有性能问题
  - Buffer pool data writes 接近 Asynchronous pool data page writes
  - Buffer pool index writes 接近 Asynchronous pool index page writes

```
$ db2 get snapshot for db on sample | grep -i write
Buffer pool data writes                = 200
Asynchronous pool data page writes     = 200
Buffer pool index writes                = 83
Asynchronous pool index page writes    = 83
```

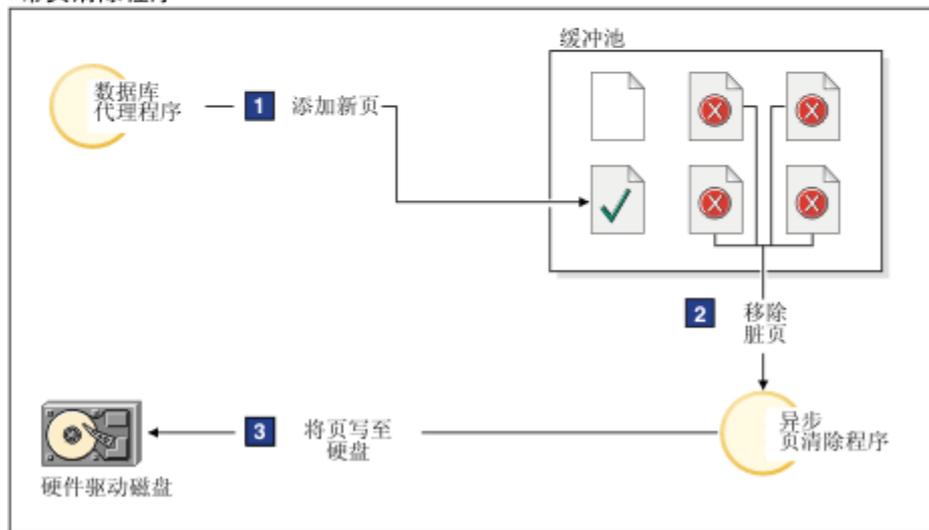
# 异步写-异步页清除过程简介

- 不带页清除程序：将 `NUM_IOCLEANERS` 参数设置为 0，缓冲池满的时候才会写磁盘
- 带页清除程序：脏页数量达到一定比例时，就会调用页清除程序，将脏页写入磁盘
  - `CHNGPGS_THRESH` 参数定义了脏页所占最大比例
  - `SOFTMAX` 参数定义了脏页涉及到的日志数

不带页清除程序



带页清除程序



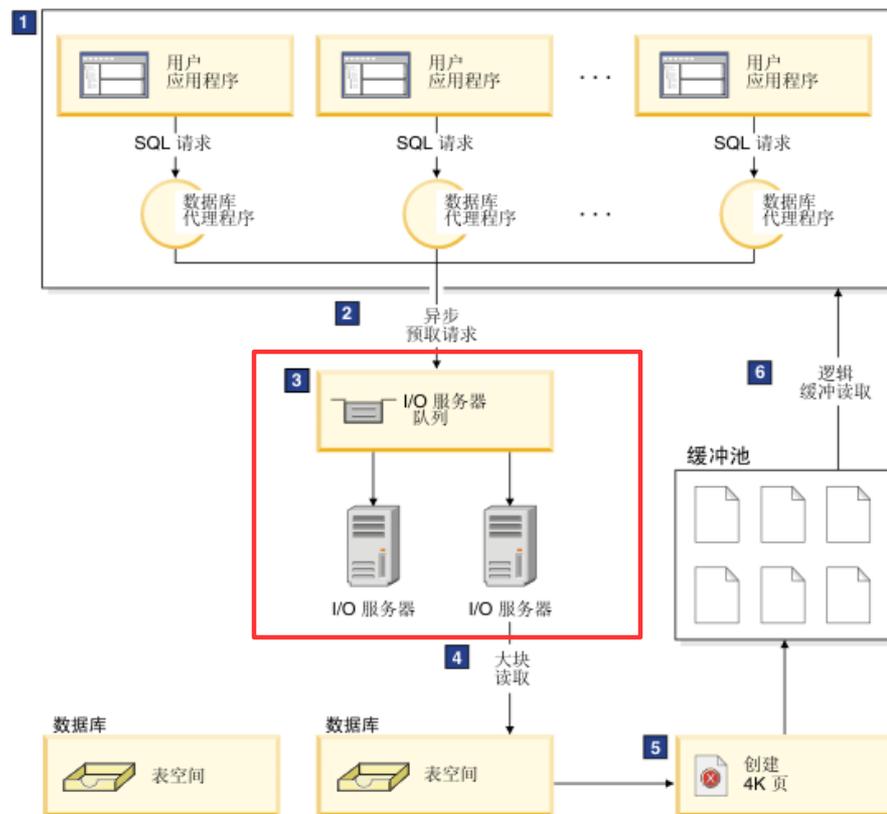
# 预取- I/O 服务器数

- NUM\_IOSERVERS参数定义了I/O服务器数
- I/O服务器的功能
  - 配合select语句执行预取(Prefetch)
  - 配合Backup/Restore等工具执行异步I/O
- NUM\_IOSERVERS参数的建议配置
  - 比物理磁盘数多一点
  - 该参数不会占用过多资源，因此可以配置稍大一些

# 预取-预取功能简介

■ 预取过程如右图所示：

1. 应用程序发送SQL到数据库代理程序
2. 如果使用预取，那么数据库代理程序将预取请求发送给I/O服务器队列
3. I/O服务器队列中的某个空闲I/O服务器处理该请求
4. I/O服务器从表空间读取大块数据(以Prefetch为单位)
5. I/O服务器将数据写入缓冲池
6. 数据库代理程序对缓冲池中的数据页执行的操作并将结果返回给应用程序



- 操作系统参数
- 注册表变量
- 数据库配置
- ➔ ■ 初始化配置
- **STMM**自动内存管理

- 运行 AUTOCONFIGURE ， 初始化配置
- 根据输入参数， 给出建议的配置

参数值	取值范围	含义
MEM_PERCENT	1-100	使用的内存占实例的比例
WORKLOAD_TYPE	simple, mixed, complex	工作负载类型
NUM_STMTS	1-1 000 000	每个事务的SQL数量
TPM	1-200 000	每分钟事务数
ADMIN_PRIORITY	performance, recovery, both	优先性能或优先恢复
IS_POPULATED	yes, no	数据是否稠密
NUM_LOCAL_APPS	0-5 000	本地连接数量
NUM_REMOTE_APPS	0-5 000	远程连接数量
ISOLATION	RR, RS, CS, UR	隔离级别
BP_RESIZEABLE	yes, no	缓冲池是否可调

# AUTOCONFIGURE 命令实例

```
db2 AUTOCONFIGURE USING \
    MEM_PERCENT      60 \
    WORKLOAD_TYPE    MIXED \
    NUM_STMTS        500 \
    TPM              1000 \
    ADMIN_PRIORITY    BOTH \
    IS_POPULATED     YES \
    NUM_LOCAL_APPS    0 \
    NUM_REMOTE_APPS   20 \
    ISOLATION         RR \
    BP_RESIZEABLE    YES \
APPLY DB AND DBM
```

- 操作系统参数
- 注册表变量
- 初始化配置
- 数据库配置
- ➔ ■ **STMM自动内存管理**

# DB2内存优化- STMM

- DB2 9开始引入了一种革命性的内存优化系统，称之为内存自我调节管理系统(STMM)
- 不需要数据库管理员的干预
  - 能够感知潜在的工作负载并且根据需要来调节优化内存
  - 能够迅速的根据工作负载来调整重新分配内存
- 能够对以下内存进行自动调整
  - 实例配置参数中定义的内存
  - 数据库配置参数中定义的内存
  - 缓冲池

# 开启STMM功能的步骤

## 1. 打开自动内存管理功能开关

```
db2 update db cfg using SELF_TUNING_MEM ON
```

DB2 9.7中默认已打开

## 2. 将下列数据库参数配置为Automatic

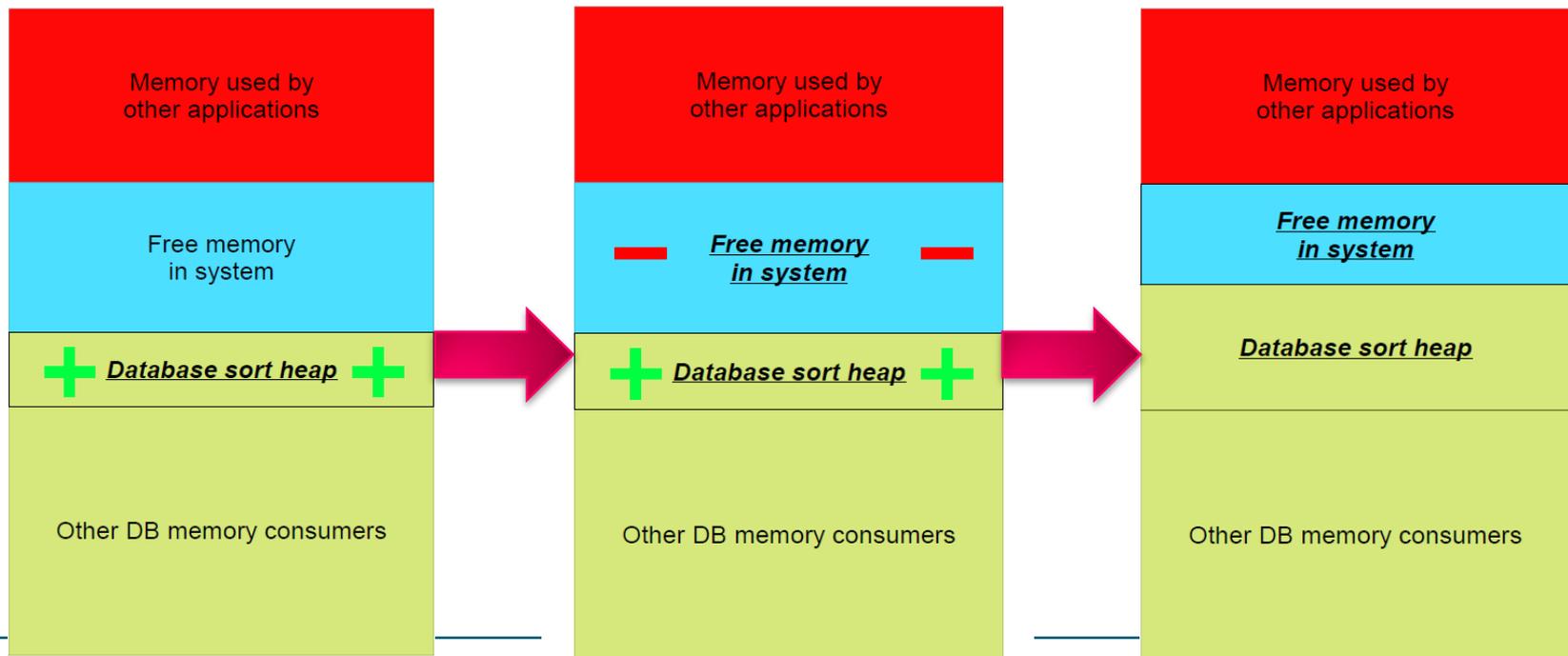
DATABASE_MEMORY
LOCKLIST
PCKCACHESZ
SHEAPTHRES_SHR
SORTHEAP
DBHEAP
STMTHEAP
APPLHEAPSZ
APPL_MEMORY
STAT_HEAP_SZ

## 3. 将bufferpool大小设置为Automatic

- 内存参数改变的一致性
  - 所有的配置变化都将写入磁盘
  - 即使重新启动或者停止数据库，所有的变化都会与之前保持一致
- 日志
  - 所有的配置变化都写入 `db2diag.log` 日志文件
  - 内部优化信息写入 `STMM.log` 日志文件
- 可选择自动调整的内存范围
  - 可将参数设置为具体数值，从而取消该参数的自动调整

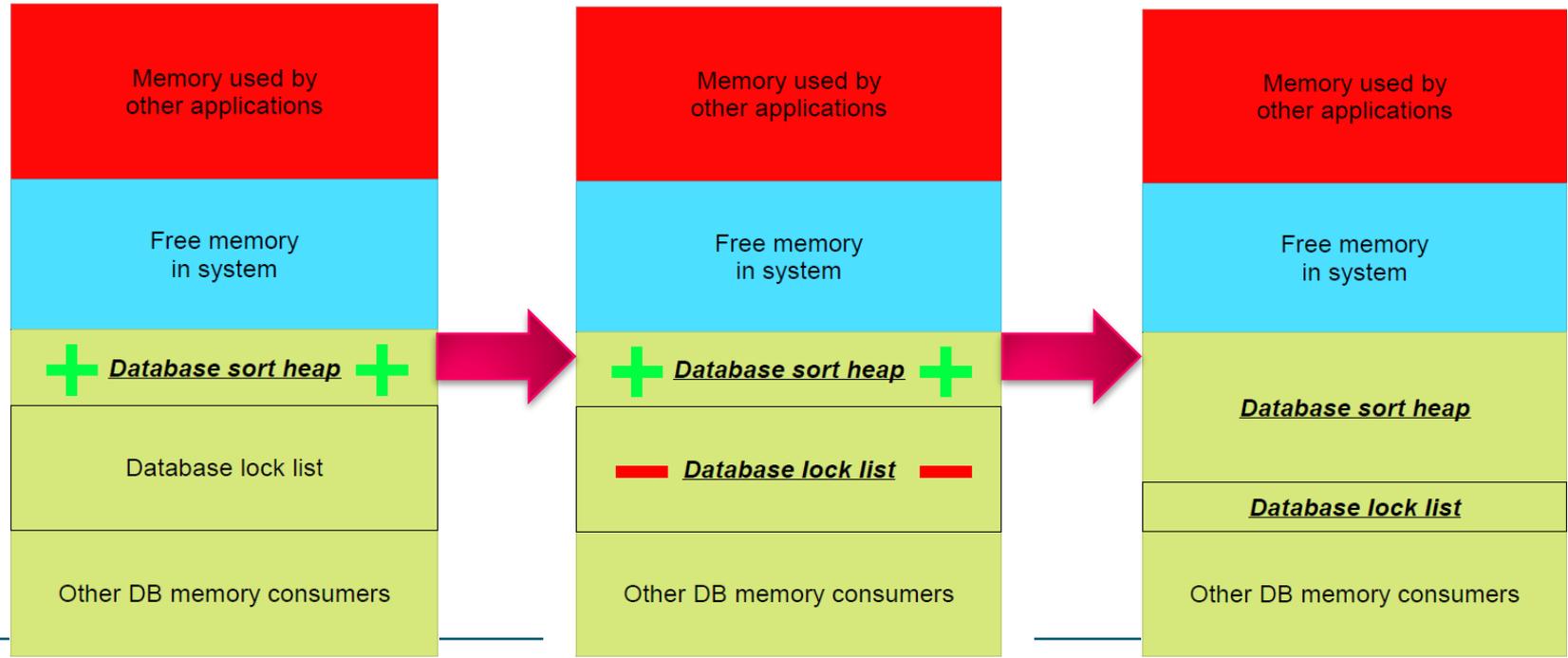
# STMM运行模式(1) – 自动数据库内存

- DATABASE\_MEMORY=AUTOMATIC
  - 可以与操作系统交换空闲内存
  - 数据库总内存可以上下浮动



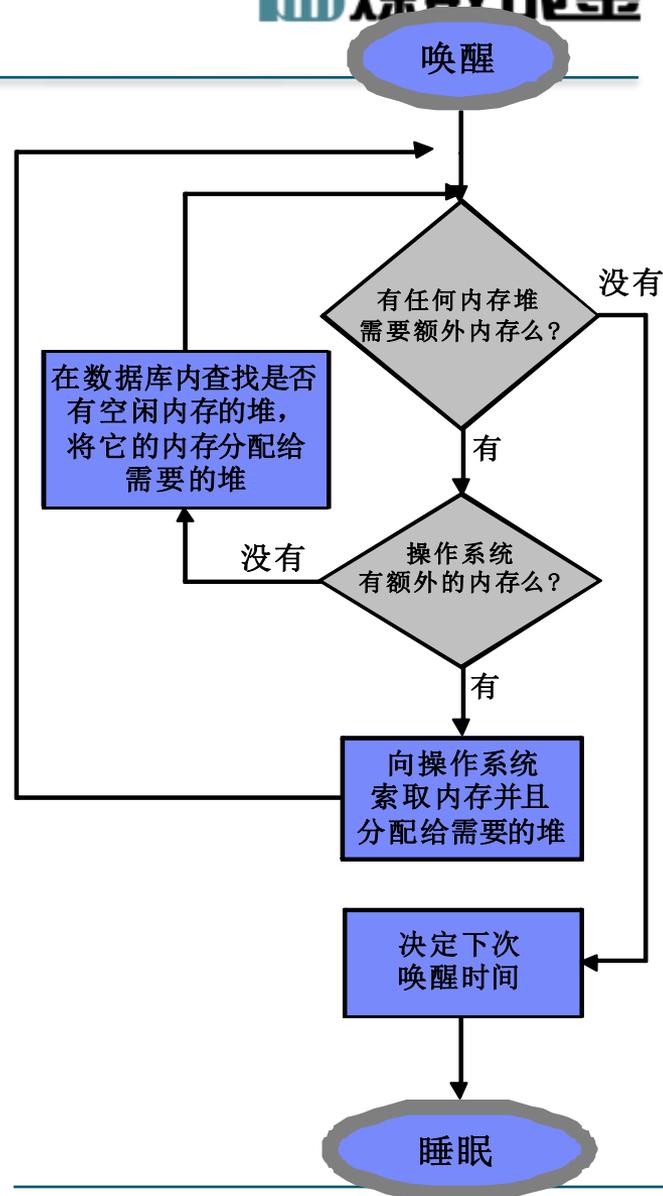
# STMM运行模式(2) – 固定数据库内存

- DATABASE\_MEMORY= 固定值或COMPUTED
  - 数据库总内存固定不变
  - 数据库内部不同内存区域可以相互交换内存



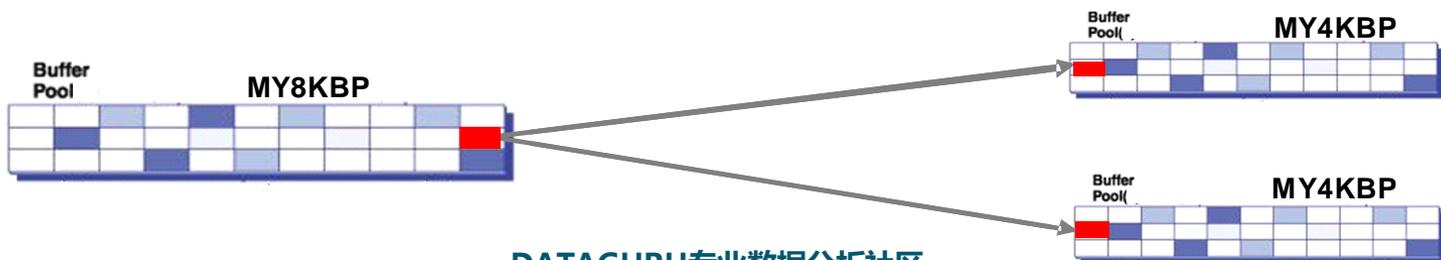
# STMM工作流程

1. STMM从休眠中醒来
2. 检查内存配置是否需要优化
3. 如果DATABASE\_MEMORY参数设置为自动
  - 从操作系统获取空闲的内存
4. 如果DATABASE\_MEMORY参数设置为固定值或者操作系统没有空闲内存
  - 从有空闲内存的堆分配空间给需要的堆
5. 继续优化直到没有可以分配的内存
6. 根据工作负载情况决定优化频率



# STMM和缓冲池

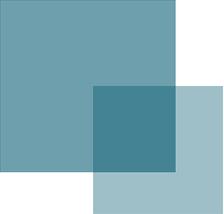
- STMM在缓冲池优化方面的优势
  - 可以平衡多个缓冲池的内存大小
  - 可以在不同页大小的缓冲池之间交换内存
  - 多个缓冲池压力不均衡时，STMM将发挥优势
  - 达到优化配置后，STMM会自动停止，以节省资源



- 操作系统参数
- 注册表变量
- 初始化配置
- 数据库配置
- **STMM**自动内存管理

Thank  
YOU

- **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