

DB2 db2top

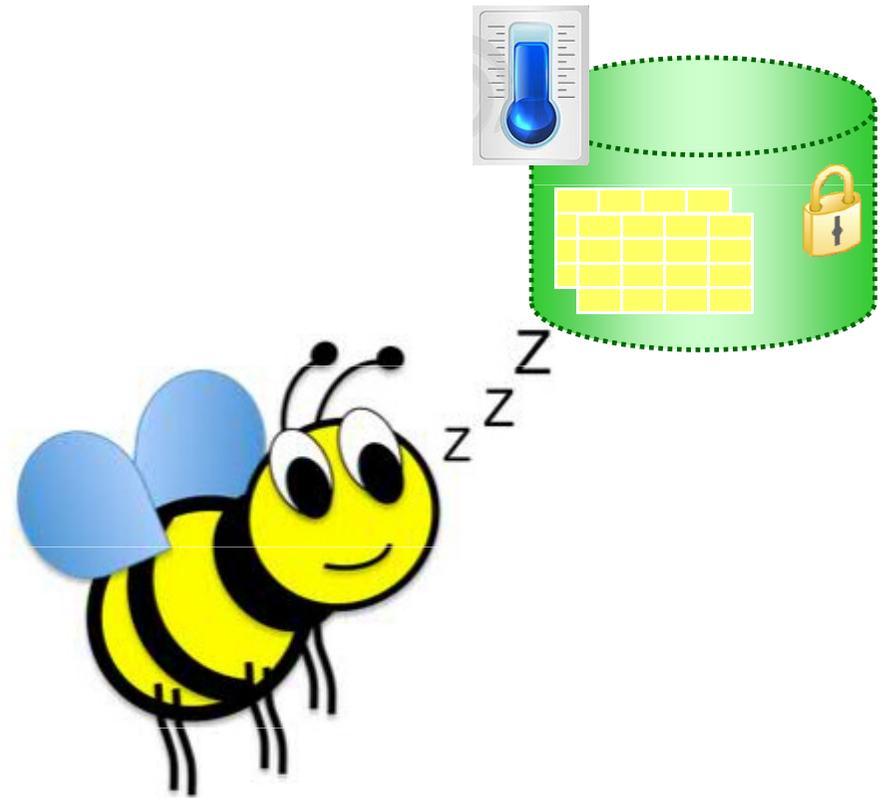
- 简单介绍及使用

Prepared by Lin Hong Feb 2016

Contact with me via 88322511@qq.com

Agent

- DB2性能分析方法
- db2top的概要
- 分析方法的简单比较
- db2top的启动
- db2top的操作
- db2top信息过滤操作
- db2top查看Utilities的进度
- 抓取执行最久执行时间SQL的执行计划
- db2top查看内存使用情况
- db2top查看Bufferpool命中率
- 通过db2top分析Lockwait状况
- db2top 历史信息收集和重放
- (参考) db2top的其他功能



DB2性能分析方法

1 收集信息 (输出到文件)

Snapshot, MON_GET函数, db2pd, db2mtrk etc.

命令的 Options 又不熟悉

输出到哪里?

这么多工具命令中, 怎么选?

2 输出文件的过滤 (数据加工)

排序, 过滤, 汇总 etc.

输出文件传输到本地

文本分析?
Excel分析?

3 输出信息的分析和诊断

db2expln, db2exfmt诊断信息的分析

1, 2, 3的步骤

命令的 Options 又不熟悉

db2top 的概要

1. GUI动态画面显示

- ❑ 针对数据库会话形式获取 snapshot 信息
- ❑ 类似nmon/top的监视

2. 可以把log信息处理到CSV文件

- ❑ 后台运行模式(-b 选项)
- ❑ 指定间隔(-i),指定期间(-m)以及可以指定回数(-s)

其他手段方法的比较

抓取最消耗时间的SQL

	方法① db2top	方法② Snapshot 命令	方法③ MON_GET 函数	方法④ 通过 db2audit
处理概要	通过db2top 可以交互式，实时地收集，过滤，分析sql执行情况	通过Snapshot命令收集信息出力出执行缓慢的sql	打开开关，通过MON_GET函数把存在内存上的监控信息处输出到csv等文件形式进行分析	通过db2audit 监视功能，利用EXECUTE catalog 抽取执行过的SQL语句
操作容易度	⊙ 不需要各种命令的使用	○ 各种option收集各种对应信息，另外需要打开monitor 开关	△ 各种表函数和视图需要很好的理解，并且需要DB cfg配置开关	△ 需要提前架构和设置，输出信息还需要进行Format
数据新鲜度	⊙ 最新信息取得，新鲜度高	○ 执行命令时候的状态信息	○ 执行命令时候的状态信息 可以定制MONREPORT	○ 执行命令时候的状态信息
数据量	○ 实时取得Snapshot信息，动态排序，并可以Explain输出执行计划	△ 只取Snapshot指定信息	⊙ 动态静态sql信息都可以收集 执行总计时间，等待时间和DB之外的等待信息也可以收集	○ 动态sql语句之外的sql，parameter mark内容，用户和执行时间可以收集到，但性能相关信息没有
使用	△ 将来强化	⊙ 从DB2 V7开始就有，经典适用于各种场景	△ 从DB2 V9.7开始提供，将来强化	△ 通常用于监视和审计使用
约束	虽然Windows环境不能使用，但可以通过做catalog连接方式进行使用			

db2top 的概要

- db2top的启动
- ✓ 不需要特殊Feature
- ✓ 是DB2 V8.1 FP17, V9.1 FP6, V9.5 FP2, V9.7 GA之后的附属功能
- ✓ 可以在AIX、Linux、HP-UX、Solaris环境下使用

(Windows除外)



- 指定DB名启动

```
$ db2top -d [DB名]
```

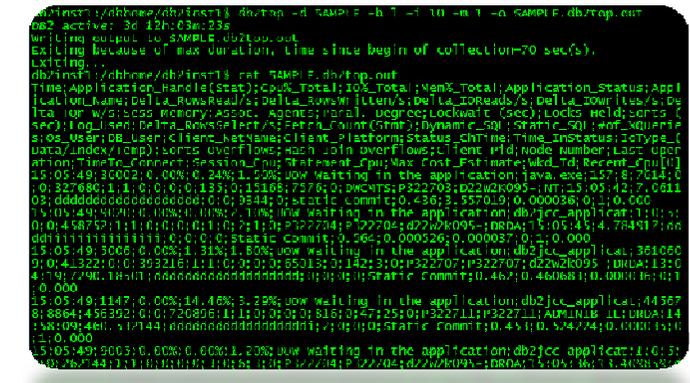
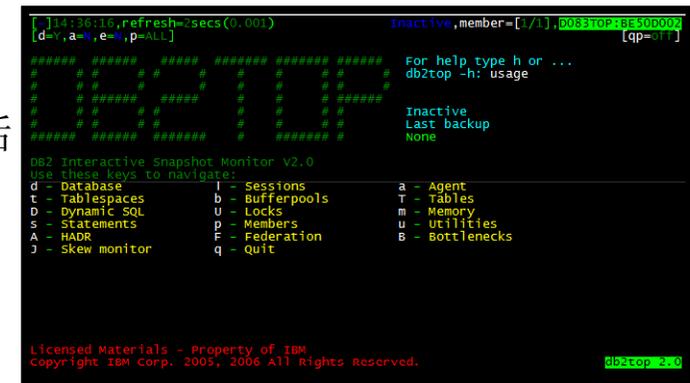
※即使错用db2stop,如果有连接存在的数据库,是停止不了实例

- 把监视信息输出到csv文件

```
$ db2top -d [DB名] -b | -i 10 -m 6 -o [文件名]
```

※指定数据库名, -b 后台模式运行

※指定间隔 (-i) 秒/ 期间(-m) 分钟/ 回数(-s)



db2top 的操作

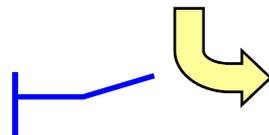
- db2top的启动后的Main菜单
- ✓ 输入字母可以进入子菜单画面

```
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 - Members          u - Utilities
A - HADR              F - Federation       B - Bottlenecks
J - Skew monitor     q - Quit
```

- ✓ 进入动态SQL子菜单

```
[V]15:14:13,refresh=1secs(0.001)      AIX,member=[ /1 ],D227TPM:TPMSDB
[d=Y,a=N,e=N,p=ALL]                      [qp=off]
#####
##### For help type h or ...
##### db2top -h: usage
#####
##### Status: Active
##### Uptime: 3d 12h:11m:47s
##### Last Backup: 2016/01/29 - 12:00:17
#####
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 - Members          u - Utilities
A - HADR              F - Federation       B - Bottlenecks
J - Skew monitor     q - Quit
Licensed Materials - Property of IBM
Copyright IBM Corp. 2005, 2006 All Rights Reserved.
db2top v2.0
```

输入 "D"



```
[V]15:15:23,refresh=1secs(0.001)      SQL      AIX,member=[ /1 ],D227TPM:TPMSDB
[d=Y,a=N,e=N,p=ALL]                      [qp=off]
SQL_Statement                               Num      Exec
HashValue                                   Execution Time
-----
00000000001143092359292996 UPDATE TTMODULE SET ISPUBLIC =      1      0.005633
000000000001973693970859476 SELECT TTCOSTESTIMATE.id, TTCO      1      0.004072
000000000003071721821800350 select tc2id.code as pid,         1      0.015700
000000000007621151190881130 SELECT FTLROLEID, FTLEDGID, CO      2      0.002004
000000000007966597916417415 SELECT uow.projectId as pid       1      0.015963
000000000009337986116815816 SELECT BEL.TTLTICKLERS.ID, TIC      52     0.177193
00000000010463661892100117 insert into JPN.ttcontact(orgi     1      0.073737
00000000012663148101485186 select 1 from CENTRAL.TLMODULE     1      0.002611
00000000013485291380952555 select ftchargetoid as c2id, f     4      0.014539
00000000013678385639730152 SELECT uow.projectId as pid       1      0.007331
00000000015877286118194457 select ftchargetoid as c2id, f     5      0.015405
00000000018116658817613573 SELECT f.projectId as pid, -      1      0.005107
00000000018405077428396634 SELECT uow.projectId as pid, -      1      0.015427
00000000018643123556037124 SELECT TTCOMMENT.ID, TTCOMMENT     1      0.011010
00000000020649210691184846 SELECT uow.projectId as pid       1      0.017737
00000000022614724077440146 SELECT uow.projectId as pid       1      0.124901
00000000023089171287712927 UPDATE TLPLANNEDVOLUME SET COP     1      0.003366
00000000026009398309116439 update JPN.ttinfounit set ftpe     1      0.032057
00000000027855815098329260 select v.changedObjectKey as c     1      0.013079
00000000028000886742528936 SELECT COUNT(UOW.ID) FRO         2      0.035402
00000000029513515146306738 SELECT f.projectId as pid, -      1      0.003732
Quit: q, Help: h      Dynamic SQL 5796 (Cached=5796), L: Query Text
db2top v2.0
```

db2top 的操作

- 监控时间的调整
- ✓ 输入“ I ” 字母，再输入数字可以调整监控画面刷新的间隔时间 (2秒更改为10秒间隔)

1. 输入 [I]

2. 输入间隔(秒)

Please enter new refresh interval: 10

3. 已更改为10秒

MaxActSess	-	25%	75%	100%
SortMemory	-			
LogUsed	-			
FCM BufLow	-			

Sessi	Start Date	Start Time	Status	Sthres	Buffers	FCMbuf	OtherMem
L_Re	2016/01/31	03:02:26	Active	0	1.6G	832.0K	878.8M
24,	Sessions	ActSess	LockUsed	LockEscals	Deadlocks	LogReads	LogWrites
	48	1	0%	0	0	0	0
Sort	L_Reads	P_Reads	HitRatio	A_Reads	writes	A_writes	Lock wait
128	5,016	2,080	58.52%	99.59%	0	0	0
	Sortheap	Sortovf	PctSortovf	AvgPRdTime	AvgDRdTime	AvgPwrTime	AvgDwrTime
	6.6M	0	0.00%	0.15	0.00	0.00	0.00

Quit: c, h

Quit: c, Help: h

allocated/sort: 1 (0)

db2top 2.0

db2top 的操作

- 监控画面的移动(左右)
- ✓ 通过“←”和“→”来调整左右两边未显示的监控项目(上下通过过滤来显示)

输入“→”

Hit Ratio

Tablespace Name	Hit Ratio%	Async Reads%	Pages Ahead	Actual writes	Actual a_reads	Actual a_writes
SCENTRAL	99.39%	98.57%	25	4,896	29,494,819	4,407
SAPLAB	66.76%	88.26%	24	0	3,158	0
SCFM	33.33%	0.00%	0	0	0	0
SYSTOOLSPACE	33.33%	0.00%	0	0	0	0
TPMSTEMPTS16	0.00%	0.00%	0	0	0	0
TPMSTEMPTS32	100.00%	0.00%	0	0	0	0
TPMSTEMPTS4	100.00%	14.39%	1	4,797	20	4,552
TSASNAA	0.00%	0.00%	0	0	0	0
TSASNCA	0.00%	0.00%	0	0	0	0
TSASNUOW	0.00%	0.00%	0	0	0	0
USERSPACE1	99.21%	36.47%	17	134	2,039	134
USERTEMP1	0.00%	0.00%	0	0	0	0
SLAB16	99.98%	41.67%	6	106	1,318	104
SNALAB	15.38%	99.75%	120	762	18,690,535	458
SYSCATSPACE	98.03%	51.47%	52	10,004	13,940	8,778
SEMEALAB	22.74%	99.15%	124	217	9,633,017	201

quit: q, Help: h DB Size 59.8G/86.2G db2top 2.0

输入“←”

db2top 的操作

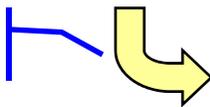
- 从监控画面获取当前信息的Native Snapshot信息
- ✓ 输入 “S” 来获取当前信息抓取的Snapshot快照信息

```

[-] 6:22:18,refresh=0secs(0.001)      SQL      AIX,member=[1/1],D227TPM:TPMSDB
[d=Y,a=N,e=N,p=ALL]                    [qp=off]

SQL_Statement      Sql      Num      Exec
HashValue          Statement (30 first char ) Execution  Time
-----
get snapshot for dynamic sql on TPMSDB global
Dynamic SQL Snapshot Result
Database name      = TPMSDB
Database path     = /db/d227tpm/d227tpm/NODE0000/SQL00001/MEMBER0000/
Number of executions      = 1
Number of compilations    = 1
Worst preparation time (ms) = 2
Best preparation time (ms) = 2
Internal rows deleted     = 0
Internal rows inserted   = 0
Rows read                 = 3
Internal rows updated     = 0
Rows written              = 0
Statement sorts           = 1
Statement sort overflows  = 0
Total sort time           = 0
Buffer pool data logical reads = 3
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 = 5
Buffer pool index physical reads = 0
"/tmp/snapshot.29360134" 279421 lines, 12724608 characters
quit: q, Help: ?
  
```

输入 “S”



db2top 信息的过滤操作

- db2top 监控画面信息可以通过正则表达式字符来过滤想要的信息，也就是上下过滤信息
- ✓ db2top 使用扩张 POSIX 正则表达式（参考如下表）

字符	含义
^	行首
\$	行尾
	左右都匹配 (or)
.(英文句号)	任意一个字符
+	前面字符多一个匹配
*(星号)	0个字符以上字符串的匹配
?	前面字符0个或者1个匹配
¥	转意字符
()	文字列处理
[]	括号内任意匹配

db2top 信息的过滤操作

- 动态SQL语句的过滤-任意一个字符的过滤 (.)
- ✓ 从Main画面按“ D” 进入动态SQL监控画面

```

[~] 07:02:41, refresh=5secs(0.002) SQL AIX, member=[1/1], D2300BM;OIOPROD
[d=y, a=N, e=N, p=ALL] [qp=off]

```

SQL_Statement HashValue	Sql Statement (30 first char.)	Num Execution	Exec Time	Avg ExecTime	Cpu Time	Avg CpuTime
00000000175938877167710863	update sessions set username =	49	0.012305	0.000251	0.007263	0.000148
00000000856998019034948202	SELECT CONTRACT_NO_NAME, PRICE_	129	0.568308	0.004405	0.062478	0.000484
00000004428622141397140338	select 1 from syscat.tables wh	5	0.099793	0.019958	0.005541	0.001108
00000005758090186561200729	UPDATE D2300BM.M_CONTRACT_DETA	20	0.020331	0.001016	0.006203	0.000310
00000006517806831225475177	SELECT CONTRACT_NO_NAME, PRICE_	46	0.013794	0.000299	0.008051	0.000175
00000008169694210992457663	update sessions set lastaccess	1,163	0.259692	0.000223	0.149656	0.000128
00000008695748045560061281	SELECT PARENT_NO, PRINT_F, SORT,	67	0.111386	0.001662	0.011905	0.000177
00000009658464535586918760	select id, lastaccess, creation	962	0.157992	0.000164	0.086776	0.000090
00000010201048167446782319	UPDATE D2300BM.T_CUSTOMER SET	2	0.003013	0.001506	0.001820	0.000910
00000010418070443567374479	SELECT U.EMP_SERIAL, U.EMP_PSC	479	0.194933	0.000406	0.074648	0.000155
00000010838040104918717804	insert into sessions (id, prop	331	0.269236	0.000813	0.069003	0.000208
00000011443355248385416573	select id, lastaccess, creation	962	0.135021	0.000140	0.071841	0.000074
00000011667070320277249013	SELECT CPNO, IOC_YEAR, IOC_01, IO	5	0.072004	0.014400	0.002108	0.000421
00000011912070495162234452	select lastaccess from session	962	0.208435	0.000216	0.113539	0.000118
00000012771050719128431202	SELECT CUST_NAME, START_YEAR, ST	132	0.059540	0.000451	0.020947	0.000158
00000013566319263584333157	INSERT INTO D2300BM.M_CONTRACT	2	0.003460	0.001730	0.001772	0.000886
00000013763211292500061382	select colNames from syscat.in	5	0.027692	0.005538	0.002974	0.000594
00000016173425336521219143	SELECT CUST_NAME, START_YEAR, ST	54	0.582700	0.010790	0.055971	0.001036
00000016559529245118830090	select CURRENT_CLIENT_ACCTNG,	2	0.084080	0.042040	0.002785	0.001392
00000017132866826540141483	WITH TYPEINTS (TYPEINT, COLTY	5	0.186131	0.037226	0.035051	0.007010
00000018299884850133322773	delete from sessions where id	309	0.053784	0.000174	0.031310	0.000101
00000018412272690243698062	CALL SYSIBM.SQLCOLUMNS (?, ?, ?,	5	0.246805	0.049361	0.007409	0.001481

Quit: q, Help: h Dynamic SQL 22 (Cached=22), L: Query Text db2top 2.0

高亮显示的sql语句为正在执行的SQL

db2top 信息的过滤操作

- 动态SQL语句的过滤-任意一个字符的过滤 (.)
- ✓ 输入" /D...OBM" 的过滤结果 (三个点表示匹配任意三个字符)

①输入" /D...OBM "

检索的过滤条件在此处显示

高亮显示结果为过滤后结果

通过输入"/" 返回全部结果

SQL_Statement HashValue	Sql Statement (30 first char.)	Num Execution	Exec Time	Avg ExecTime	Cpu Time	Avg CpuTime
00000001739583	update sessions set lastaccess	49	0.012305	0.000251	0.007263	0.000148
00000000856998	select id,lastaccess, creation	129	0.568308	0.004405	0.062478	0.000484
00000004428622141397140338	select id,lastaccess, creation	5	0.099793	0.019958	0.005541	0.001108
00000005758090186561200729	UPDATE D2300BM.M_CONTRACT_DETA	20	0.020331	0.001016	0.006203	0.000310
00000006517806831225475177	SELECT CONTRACT_NO_NAME,PRICE_	46	0.013794	0.000299	0.008051	0.000175
00000008169694210992457663	update sessions set lastaccess	1,165	0.260288	0.000223	0.149915	0.000128
00000008695748045560061281	SELECT PARENT_NO,PRINT_F,SORT,	67	0.111386	0.001662	0.011905	0.000177
00000009658464535586918760	select id,lastaccess, creation	964	0.158518	0.000164	0.086982	0.000090
00000010201048167446782319	UPDATE D2300BM.T_CUSTOMER SET	2	0.003013	0.001506	0.001820	0.000910
00000010418070443567374479	SELECT U.EMP_SERIAL, U.EMP_PSC	479	0.194933	0.000406	0.074648	0.000155
00000010838040104918717804	insert into sessions (id, prop	331	0.269236	0.000813	0.069003	0.000208
00000011443355248385416573	select id,lastaccess, creation	964	0.135501	0.000140	0.072039	0.000074
00000011667070320277249013	SELECT CPNO,IOC_YEAR,IOC_01,IO	5	0.072004	0.014400	0.002108	0.000421
00000011912070495162234452	select lastaccess from session	964	0.208778	0.000216	0.113722	0.000117
00000012771050719128431202	SELECT CUST_NAME,START_YEAR,ST	132	0.059540	0.000451	0.020947	0.000158
00000013566319263584333157	INSERT INTO D2300BM.M_CONTRACT	2	0.003460	0.001730	0.001772	0.000886
00000013763211292500061382	select colNames from syscat.in	5	0.020331	0.001506	0.001820	0.000910
00000016173425336521219143	SELECT CUST_NAME,START_YEAR,ST	54	0.58	0.001506	0.001820	0.000910
00000016559529245118830090	select CURRENT_CLIENT_ACCTNG,	2	0.08	0.001506	0.001820	0.000910
00000017132866826540141483	WITH TYPEINTS (TYPEINT, COLTY	5	0.186131	0.037226	0.035051	0.007010

db2top 信息的过滤操作

- 动态SQL语句的过滤-转移字符的过滤 (\)
- ✓ 输入" ^?" 的过滤结果 (过滤有 ? 的语句)

```

[-]17:19:11,refresh=5secs(0.002)      SQL      AIX,member=[1/1],D2300BM:OIOPRODD
[d=Y,a=N,e=N,p=ALL]                    [qp=off]
^?
-----
SQL_Statement      Ssql      Num      Exec      Avg      Cpu      Avg
HashValue          Statement (30 first char.) Execution Time  ExecTime  Time     CpuTime
-----
00000008168093    access      1,199    0.266563  0.000222  0.153619  0.000128
00000009658464    creation    998      0.168258  0.000168  0.090054  0.000090
0000001144335524838341833 select id,lastaccess, creation
00000011912070495162234452 select lastaccess from session
0000000175938877167710863 update sessions set username =
00000000856998019034948202 SELECT CONTRACT_NO_NAME,PRICE_
00000004428622141397140338 select i from syscat.tables wh
00000005758090186561200729 UPDATE D2300BM.M_CONTRACT_DETA
00000006517806831225475177 SELECT CONTRACT_NO_NAME,PRICE_
00000008695748045560061281 SELECT PARENT_NO,PRINT_F,SORT,
00000010201048167446782319 UPDATE D2300BM.T_CUSTOMER SET
00000010418070443567374479 SELECT U.EMP_SERIAL, U.EMP_PSC
00000010838040104918717804 insert into sessions (id,prop
00000011667070320277249013 SELECT CPNO,IOC_YEAR,IOC_01,IO
00000012771050719128431202 SELECT CUST_NAME,START_YEAR,ST
00000013566319263584333157 INSERT INTO D2300BM.M_CONTRACT
00000013763211292500061382 select colNames from syscat.in
00000016173425336521219143 SELECT CUST_NAME,START_YEAR,ST
00000016559529245118830090 select CURRENT_CLIENT_ACCTNG,
00000017132866826540141483 WITH TYPEINTS ( TYPEINT, COLTY
000000182008185012222777 delete from sessions where id
0000 [7]17:21:05,refresh=114!secs(0.002)  SQL      AIX,member=[1/1],D2300BM:OIOPRODD
[d=Y,a=N,e=N,p=ALL]                    [/?',(p=off)]
-----
SQL_Statement      Ssql      Num      Exec      Avg      Cpu      Avg
Hashvalue          Statement (30 first char.) Execution Time  ExecTime  Time     CpuTime
-----
00000018412272690243698062 CALL SYSIBM.SQLCOLUMNS (?,?,?

```

高亮显示结果为过滤后结果

通过输入" /" 返回全部结果

db2top 信息的过滤操作

- 动态SQL语句的过滤-多一个字符以上匹配 (+)
- ✓ 输入“ /SYSINSTAL+ ” 的过滤结果 (过滤有SYSINSTAL,SYSINSTALL等的SQL语句)

检索的过滤条件在此处显示

①输入“ /SYSINSTAL+ ”

SQL_Statement HashValue	Sql Statement (30 first char.)	Num Execution	Exec Time
00000005429283171301468277	CALL SYSPROC.SYSINSTALLOBJECTS	33,235	75.666435
00000008161574099279669012	CALL SYSPROC.SYSINSTALLOBJECTS('DB2AC	946	7.952181

SYSINSTAL, SYSINSTALL, SYSINSTALLL等过滤后结果

通过输入“ / ” 返回全部结果

db2top 信息的过滤操作

- 动态SQL语句的过滤-输出列的显示
- ✓ 输入“ c ” 然后输入想要显示的列 (把想显示的列显示出来)

SQL Statement HashValue

Sql Statement (30 first char.)	Num Execution	Exec Time	Avg ExecTime	Cpu Time	Avg CpuTime
update sessions set username =	51	0.012	0.000251	0.007516	0.000147
SELECT CONTRACT_NO_NAME,PRICE	134	0.004362	0.004362	0.065018	0.000485

SQL

Pos	Default	Actual	Sort
0	SQL_Statement Hashvalue	SQL_Statement Hashvalue	Asc
1	Sql Statement	Sql Statement	-
2	Num Execution	Num Execution	-
3	Exec Time	Exec Time	-
4	Avg ExecTime	Avg ExecTime	-
5	Cpu Time	Cpu Time	-

POS : 列号
Default : 默认排序 (列名)
Actual : 现在实际排的列名
Sort : 升序还是降序
如果想修改默认排序需要修改 .db2toprc 文件

SQL

SQL_Statement HashValue	Exec Time	Data Hit%	Index Hit%	Temp Hit%	Sql Statement (30 first char.)	Num Execution	Avg ExecTime
00000008169694210992457663	0.278354	100.00%	100.00%	0.00%	update sessions set lastaccess	1,231	0.000222
00000009658464535586918760	0.175802	100.00%	100.00%	0.00%	select id,lastaccess, creation	1,041	0.000168
00000011443355248385416573	0.145619	100.00%	100.00%	0.00%	select id,lastaccess, creation	1,041	0.000139
00000011912070495162234452	0.225651	100.00%	100.00%	0.00%	select lastaccess from session	1,041	0.000216
0000000175938877167710863	0.012832	100.00%	100.00%	0.00%	update sessions set username =	51	0.000251
00000000856998019034948202	0.584544	100.00%	100.00%	0.00%	SELECT CONTRACT_NO_NAME,PRICE	134	0.004362
00000004428622141397140338	0.099793	100.00%	100.00%	0.00%	select 1 from syscat.tables wh	5	0.019958
00000005758090186561200729	0.020664	100.00%	100.00%	0.00%	UPDATE D2300BM.M_CONTRACT_DETA	21	0.000984
00000006517806831225475177	0.014269	100.00%	100.00%	0.00%	SELECT CONTRACT_NO_NAME,PRICE	48	0.000297
00000008695748045560061281	0.112068	100.00%	100.00%	0.00%	SELECT PARENT_NO,PRINT_F,SORT,	70	0.001600
00000010201048167446782319	0.003013	100.00%	100.00%	0.00%	UPDATE D2300BM.T_CUSTOMER SET	2	0.001506
00000010418070443567374479	0.199955	100.00%	100.00%	0.00%	SELECT U.EMP_SERIAL, U.EMP_PSC	498	0.000401
00000010838040104918717804	0.273313	100.00%	100.00%	0.00%	insert into sessions (id,prop	343	0.000796
00000011667070320277249013	0.072004	100.00%	100.00%	0.00%	SELECT CPNO,IOC_YEAR,IOC_01,IO	5	0.014400
00000012771050719128431202	0.060827	100.00%	100.00%	0.00%	SELECT CUST_NAME,START_YEAR,ST	137	0.000443
00000013566319263584333157	0.003460	100.00%	100.00%	0.00%	INSERT INTO D2300BM.M_CONTRACT	2	0.001730
00000013763211292500061382	0.027692	100.00%	100.00%	0.00%	select ColNames from syscat.in	5	0.005538
0000001617342533652121914	0.000000	100.00%	100.00%	0.00%	SELECT CUST_NAME,START_YEAR,ST	56	0.010642
0000001655952924511883009	0.000000	100.00%	100.00%	0.00%	select CURRENT CLIENT_ACCTNG,	2	0.042040
0000001713286682654014148	0.000000	100.00%	100.00%	0.00%	WITH TYPEINITS (TYPEINT, COLTY	5	0.037226
0000001829988485013332277	0.000000	100.00%	100.00%	0.00%	delete from sessions where id	331	0.000174
0000001841227269924369806	0.000000	100.00%	100.00%	0.00%	CALL SYSIBM.SQLCOLUMNS (?,?,?	5	0.049361

①输入“ c ”
②输入列号码
③Enter

dynsql=
Enter new columns order for dynsql: 3,10,12,14
Quit: q, Help: h

Press enter to resume

db2top 2.0

db2top 查看Utilities的进度

- 输入小写“u”即可查看当前utilities工具的执行进度

```
[~]00:57:12,refresh=2secs(0.001) utilities AIX,member=[1/1] 201602151550001
[d=Y,a=N,e=N,p=ALL]
Hash Value # of entries Utility Start Time Utility Type Uti Pri State Invoker Completed Work Unit Prog%
-----
2122241 1 00:55:21.737794 Backup 0 Execute User 775.4M Bytes 49%
```

输入“u”

当前Backup正在执行进度是49%

```
Quit: q, Help: h g19aedrdb101 db2top 2.0
```

抓取执行最久执行时间SQL的执行计划

- 一般情况，我们可以通过snapshot & EXPLAIN 来抓取执行最久的sql，并解析期执行计划，步骤有：抓取snapshot后分析出执行最久SQL语句，在explain分析获取执行计划；而通过db2top则可以方便地实时地去获取并解析SQL的执行计划
- 通过db2top的解析步骤：

Step1

定位找出平均执行时间最大的SQL

Step2

SQL内容的确认

Step3

获取SQL的执行计划

Step1: 定位找出平均执行时间最大的SQL

- Dynamic SQL监控画面中，按照平均执行时间排序
 - 输入小写“z”并输入想排序的列号，进行降序排序（升序：大写“Z”）
 - 平均执行时间（AvgExecTime）默认是第4列，所以输入“z”和“4”进行降序排列

column number for descending sort: 4

SQL_Statement Hashvalue	SQL Statement (30 first char.)	Num Execution	Exec Time	Avg ExecTime	Cpu Time	Avg CpuTime
0000001841227260243698062	CALL SYSIBM.SQLCOLUMNS (?, ?, ?, ?,	5	0.247602	0.049520	0.007890	0.001578
0000001655952924118830	IENT_ACCTNG,	2	0.084080	0.042040	0.002785	0.001392
000000171328668264014	YPEINT, COI TY	0	0.186131	0.037226	0.035051	0.007010
00000004428622141397140	cat.table	0	0.099793	0.019958	0.005541	0.001108
00000011667070320277244	EAR_FOC_0	0	0.072004	0.014400	0.002108	0.000421
00000016173425336521210	START_YEAI	0	0.596001	0.010642	0.057964	0.001035
0000001376321129250006	rom syscat ...	5	0.027692	0.005538	0.002974	0.000594
00000000856998019034948	O_NAME,PRICE	34	0.584544	0.004362	0.065018	0.000485
0000001356631926358433	BM.M_CONTRACT	2	0.003460	0.001730	0.001772	0.000886
00000008695748045560061281	SELECT PARENT_NO,PRINT_F, SORT,	70	0.112068	0.001600	0.012296	0.000175
00000010201048167446782319	UPDATE D2300BM.T_CUSTOMER SET	2	0.003013	0.001506	0.001820	0.000910
00000000531126470104823957	VALUES CAST(? AS CLOB(99))	1	0.001220	0.001220	0.000626	0.000626
00000015423400223630463817	VALUES CAST(? AS CLOB(34))	1	0.001017	0.001017	0.000564	0.000564
00000005758090186561200729	UPDATE D2300BM.M_CONTRACT_DET	21	0.020664	0.000984	0.006408	0.000305
00000010838040104918717804	insert into sessions (id, prop	343	0.273313	0.000796	0.071317	0.000207
00000012771050719128431202	SELECT CUST_NAME, START_YEAR, ST	137	0.060827	0.000443	0.021691	0.000158
00000010418070443567374479	SELECT U.EMP_SERIAL, U.EMP_PSC	498	0.199955	0.000401	0.077487	0.000155
00000006517806831225475177	SELECT CONTRACT_NO_NAME, PRICE_	48	0.014269	0.000297	0.008326	0.000173
00000000175938877167710863	update sessions set username =	51	0.012832	0.000251	0.007516	0.000147
00000008169694210992457663	update sessions set lastaccess	1,367	0.303153	0.000221	0.174535	0.000127
00000011912070495162234452	select lastaccess from session	1,157	0.251895	0.000217	0.136751	0.000118
00000018299884850133322773	delete from sessions where id	343	0.059623	0.000173	0.034649	0.000101
00000009658464535586918760	select id,lastaccess, creation	1,157	0.194998	0.000168	0.104271	0.000090
00000011443355248385416573	select id,lastaccess, creation	1,157	0.159603	0.000137	0.085027	0.000073
00000007272678963134653650	VALUES :H00002 INTO :H00001	1	0.000000	0.000000	0.000000	0.000000
00000008525963133611800025	DECLARE EXTRACTSECTUR CURSOR	1	0.000000	0.000000	0.000000	0.000000
00000009421191188454387397	VALUES LENGTH(:H00007) INTO :H	1	0.000000	0.000000	0.000000	0.000000
00000011478395408671704538	CALL sysproc.exdyn_dynamic(:H0	1	0.000000	0.000000	0.000000	0.000000
00000015120190962058338241	VALUES (1) INTO :H00001	1	0.000000	0.000000	0.000000	0.000000
00000015120246821440963806	VALUES (1) INTO :H00004	1	0.000000	0.000000	0.000000	0.000000
00000016513581478900428993	VALUES SUBSTR(:H00007, :H00009	1	0.000000	0.000000	0.000000	0.000000
00000016851842101523185638	VALUES :H00006 INTO :H00007	1	0.000000	0.000000	0.000000	0.000000
00000016925240160768021257	VALUES :H00002 INTO :H00003	1	0.000000	0.000000	0.000000	0.000000

Quit: q, Help: h

Dynamic SQL 33 (Cached=33), L: Query Text

db2top 2.0

Step2: SQL内容的确认

■ SQL内容的获取

- 输入大写“L”并输入想获取SQL的Hashvalue（拷贝黏贴），即可获取整条SQL语句
- 平均执行时间（AvgExecTime）已降序排序，黏贴最上面一条的Hashvalue

```

[11:37:00,refresh=1secs(0.003) SQL AIX,member=[1/1],DB2300BM:TOPROD [qp=off]
d=N,a=N,e=N,p=ALL]
Enter SQL_hash string: 00000018412272690243698062
SQL_Statement Hashvalue Num Execution Exec Time Avg ExecTime Cpu Time Avg CpuTime
-----
00000018412272690243698062 CALL SYSIBM.SQLCOLUMNS (?,?,?, 5 0.247602 0.049520 0.007890 0.001578
00000016933923249118038093 select CURRENT CLIENT_ACCTNG, 2 0.084080 0.042040 0.002785 0.001392
00000017132866826540141483 WITH TYPEINTS ( TYPEINT, COITY 0.186131 0.037226 0.035051 0.007010
000000104428622141397140338 select 1 from syscat.table 0.099793 0.019958 0.005541 0.001108
0000001667070320277249013 SELECT CPNO,IOC_YEAR,IOC_0 0.072004 0.014400 0.002108 0.000421
0000001613425336521219143 SELECT CUST_NAME,START_YEA 0.596001 0.010642 0.057964 0.001035
0000001376311292500061 5 0.027692 0.005538 0.002974 0.000594
0000000856998019054738 234 0.584544 0.004362 0.065018 0.000485
00000013566319263584333 2 0.003460 0.001730 0.001772 0.000886
00000008695748045560061 70 0.112068 0.001600 0.012296 0.000175
00000010201048167446782 2 0.003013 0.001506 0.001820 0.000910
00000000531126470104823 1 0.001220 0.001220 0.000626 0.000626
00000015423400223630463 1 0.001017 0.001017 0.000564 0.000564
00000005758090186561200 21 0.020664 0.000984 0.006408 0.000305
00000010838040104918717804 UPDATE DB2300BM.CONTRACT_BETA 343 0.273313 0.000796 0.071317 0.000207
00000012771050719128431202 insert into sessions (id,prop 137 0.060827 0.000443 0.021691 0.000158
00000010418070443567374479 SELECT U.EMP_SERIAL, U.EMP_PSC 498 0.199955 0.000401 0.077487 0.000155
00000006517806831225475177 SELECT CONTRACT_NO_NAME,PRICE 48 0.014269 0.000297 0.008326 0.000173
0000000175938877167710863 update sessions set username = 51 0.012832 0.000251 0.007516 0.000147
00000008169694210992457663 update sessions set lastaccess 1,369 0.303501 0.000221 0.174744 0.000127
00000011912070495162234452 select lastaccess from session 1,159 0.252394 0.000217 0.137021 0.000118
00000018299884850133322773 delete from sessions where id 343 0.059623 0.000173 0.034649 0.000101
00000009658464535586918760 select id,lastaccess, creation 1,159 0.195256 0.000168 0.104414 0.000090
00000011443355248385416573 select id,lastaccess, creation 1,159 0.159766 0.000137 0.085120 0.000073
00000007272678963134653650 VALUES :H00002 INTO :H00001 1 0.000000 0.000000 0.000000 0.000000
00000008525963133611800025 DECLARE EXTRACTSECTCUR CURSOR 1 0.000000 0.000000 0.000000 0.000000
00000009421191188454387397 VALUES LENGTH(:H00007) INTO :H 1 0.000000 0.000000 0.000000 0.000000
00000011478395408671704538 CALL sysproc.exdyn_dynamic(:H0 1 0.000000 0.000000 0.000000 0.000000
00000015120190962058338241 VALUES (1) INTO :H00001 1 0.000000 0.000000 0.000000 0.000000
00000015120246821440963806 VALUES (1) INTO :H00004 1 0.000000 0.000000 0.000000 0.000000
00000016513581478900428993 VALUES SUBSTR(:H00007, :H00009 1 0.000000 0.000000 0.000000 0.000000
000000168518421101523185638 VALUES :H00006 INTO :H00007 1 0.000000 0.000000 0.000000 0.000000
00000016925240160768021257 VALUES :H00002 INTO :H00003 1 0.000000 0.000000 0.000000 0.000000
quit: q, Help: ? Dynamic SQL 33 (Cached=33), L: Query Text db2top 2.0

```

Step3: SQL的执行计划

- SQL执行计划的获取
 - 输入小写“e”，即可获取平均执行时间最久SQL的执行计划

[-] 1:37:00, refresh=1secs(0.003) SQL AIX, member=[1/1], D2300BM:QIOPRODD [qp=off]

SQL_Statement	Sql	Num	Exec	Avg	Cpu	Avg
Hashv	Query text				e	CpuTime
00000	Text for query #00000018412272690243698062 [5 executions, 1 strings]				0	0.001578
00000					5	0.001392
00000					1	0.007010
00000	CALL SYSIBM.SQLCOLUMNS (?, ?, ?, ?, ?)				1	0.001108
00000					8	0.000421
00000					4	0.001035
00000					4	0.000594
00000					8	0.000485
00000					2	0.000886
00000					6	0.000175
00000					0	0.000910
00000					6	0.000626
00000					4	0.000564
00000					8	0.000305
00000					7	0.000207
00000					1	0.000158
00000					7	0.000155
00000					6	0.000173
00000					6	0.000147
00000					4	0.000127
00000					1	0.000118
00000					9	0.000101
00000					4	0.000090
00000					8	0.000073

Quit: q, Help: ?

dynamic SQL 33 (Cached=33), L: Query Text db2top 2.0

输入“e”执行db2expln
 输入“x”执行db2exfmt
 输入“w”执行保存
 输入“E”编辑SQL问(用vi打开)
 输入“E”返回

Step3: SQL的执行计划

- SQL执行计划的确认
 - vi模式输出了explain执行计划，可以输入“w 文件名” 可以保存执行计划
 - vi退出之后就回退到db2top

```
CALL SYSIBM.SQLCOLUMNS (?, ?, ?, ?, ?)

Section Code Page = 1208
Estimated Cost = 0.000021
Estimated Cardinality = 0.000000

( 2) Call Procedure
      Name = SYSIBM.SQLCOL
      Specific Name = COLUMNS
      SQL Access Level = Modifies SQL Data
      Language = C
      Parameter Style = DB2SQL
      Expected Result Sets = 1
      Not Fenced
      Called on NULL Input
      Threadsafe
      Not Deterministic
      Disallow Parallel
      Not Secured
( 1) Return Data to Application
      #Columns = 5

End of section

Optimizer Plan:

      Rows
      Operator
      (ID)
      Cost
      0
      RETURN
      ( 1)
      2.12555e-05
      |
      *
      |
      Procedure:
      SYSIBM
      SQLCOL
```

①按Esc
②输入“w” 或者
“w 文件名” 保存

:w

db2top查看内存使用情况

- db2mtrk和db2top查看内存不同处：
 - db2top汇总了实例，数据库，应用级别内存
 - db2top列出现有内存使用容量，最高水位点和最大容量
 - db2top列出bufferpool，应用级别

db2mtrk示例

```

$ db2mtrk -i -d -p
Tracking Memory on: 2016/02/22 at 23:21:42
Memory for instance
  other    fcmbp    monh
  70.2M    832.0K    704.0K

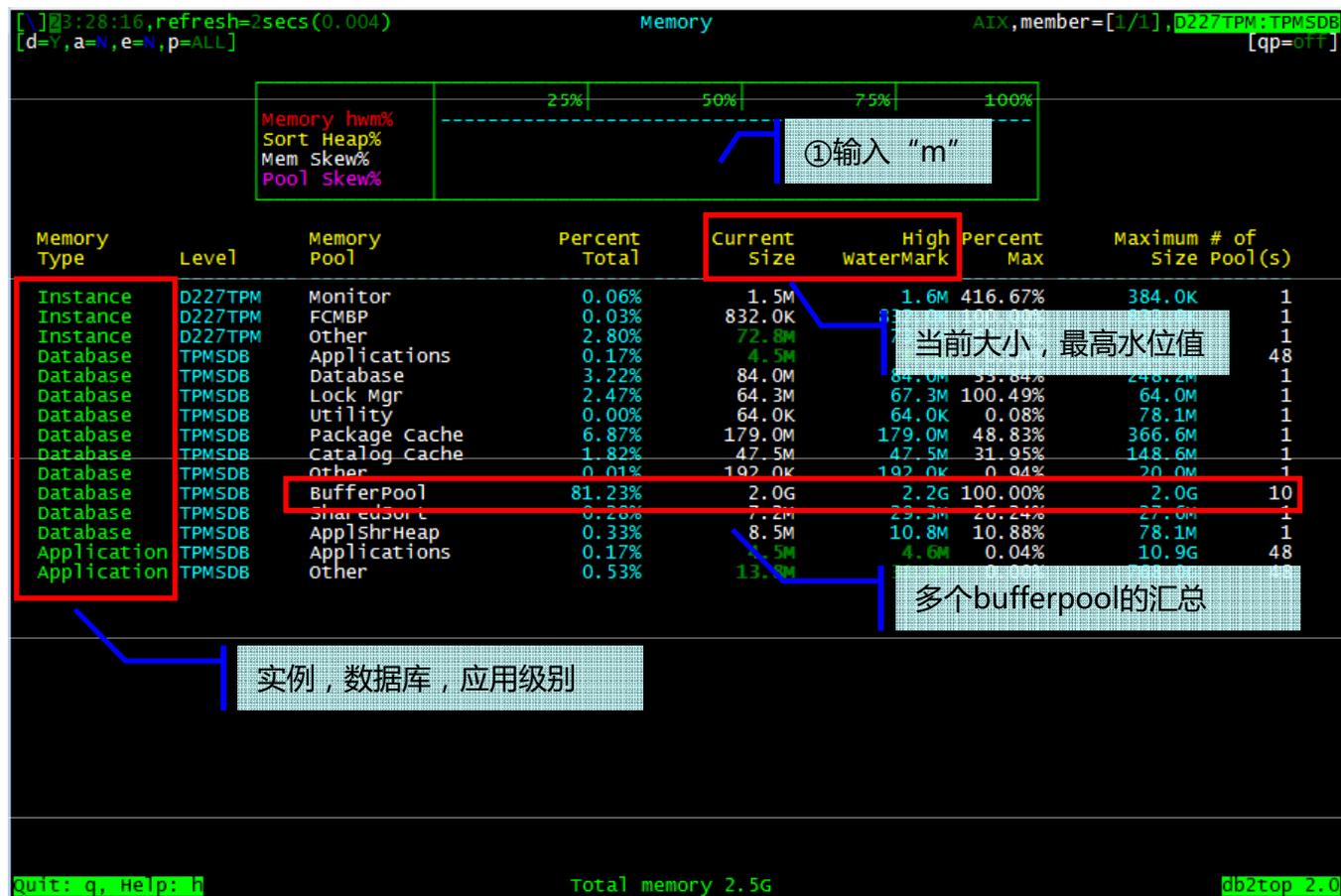
Memory for database: BE50D001
  utilh    pckcacheh  other    catcacheh  bph (2)  bph (1)
  64.0K    76.9M     192.0K    1.8M       178.5M   427.2M
  bph (S32K) bph (S16K) bph (S8K) bph (S4K) shsorth  lockh
  832.0K    576.0K    448.0K    384.0K     192.0K   179.9M
  dbh      apph (5466) apph (3575) apph (3549) apph (3273) apph (3272)
  69.1M    128.0K    128.0K    128.0K     64.0K    64.0K
  apph (3271) apph (3270) apph (3269) apph (3268) apph (3267) apph (3266)
  64.0K    64.0K     64.0K     64.0K     64.0K    64.0K

Memory for agent 3600
  other
  192.0K
.....

```

db2top查看内存使用情况

- db2top 中内存的查看
 - 输入小写“m”即可查看内存使用情况



db2top查看bufferpool命中率

- Bufferpool命中率(输入消息 "b" 即可查看)

```
[/]23:52:43,refresh=2secs(0.004) Bufferpools AIX,member=[1/1],D227TPM:TPMSDB
[d=Y,a=N,e=N,p=ALL] [qp=off]
```

Hit Ratio%	25%	50%	75%	100%				
Bufferpool Name	Delta l_reads/s	Delta p_reads/s	Hit Ratio%	Async Reads%	Delta writes/s	Delta a_reads/s	Async ReadMs	Delta a_writes/s
IBMDEFAULTBP	416	0	100.00%	0.00%	0	0	0.00	0
TC_BP	1,058	0	100.00%	0.00%	0	0	0.00	0
IBMSYSTEMBP16K	0	0	0.00%	0.00%	0	0	0.00	0
IBMSYSTEMBP32K	0	0	0.00%	0.00%	0	0	0.00	0
IBMSYSTEMBP4K	0	0	0.00%	0.00%	0	0	0.00	0
IBMSYSTEMBP8K	0	0	0.00%	0.00%	0	0	0.00	0
LAB_BP	0	0	0.00%	0.00%	0	0	0.00	0
TPMSBP16	0	0	0.00%	0.00%	0	0	0.00	0
TPMSBP32	0	0	0.00%	0.00%	0	0	0.00	0
TPMSBP8K	0	0	0.00%	0.00%	0	0	0.00	0

Quit: q, Help: h b19aedrdb1011 db2top 2.0

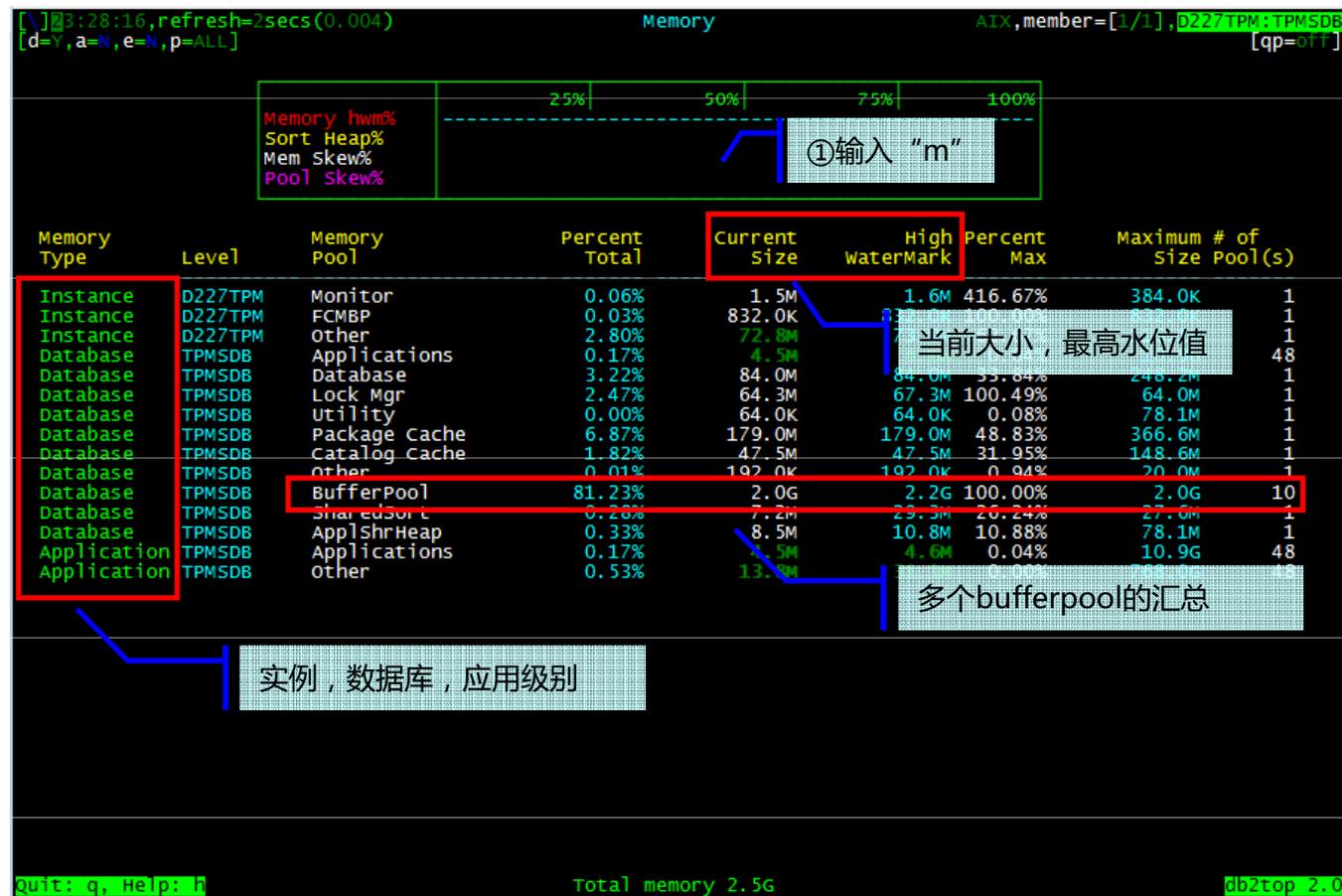
Bufferpool名

Bufferpool命中率

l_read:逻辑读 (从cache中读取)
p_read:物理读 (从磁盘中读取)

db2top查看内存使用情况

- db2top 中内存的查看
 - 输入小写“m”即可查看内存使用情况



通过db2top分析Lockwait状况

- 一般情况，我们可以通过snapshot解析需要通过snapshot获取lock情况，然后解析后，再结合获取application的snapshot，动态sql语句snapshot来分析出SQL语句; db2top可以动态解析出lockwait的agent，然后找到对应的sql语句，也还可以获取sql的执行计划
- 通过db2top的解析步骤：



Scenario脚本准备

✓ Terminate01

在DB2 9.7上，Currently Committed机制是默认打开的，这个机制会影响到死锁的模拟。

```
db2inst1:/dbhome/db2inst1$ db2 connect to sample
db2inst1:/dbhome/db2inst1$ db2 get db cfg for sample | grep -i CUR_COMMIT
Currently Committed          (CUR_COMMIT) = ON
db2inst1:/dbhome/db2inst1$ db2 update db cfg using CUR_COMMIT disabled
DB20000I The UPDATE DATABASE CONFIGURATION command completed successfully.
SQL1363W One or more of the parameters submitted for immediate modification
were not changed dynamically. For these configuration parameters, the database
must be shutdown and reactivated before the configuration parameter changes
become effective.
db2inst1:/dbhome/db2inst1$ db2 terminate
db2inst1:/dbhome/db2inst1$ db2 deactivate db sample
db2inst1:/dbhome/db2inst1$ db2 activate db sample
db2inst1:/dbhome/db2inst1$ db2 connect to sample
Database Connection Information
Database server      = DB2/AIX64 10.1.5
SQL authorization ID = DB2INST1
Local database alias = SAMPLE
db2inst1:/dbhome/db2inst1$ db2 get db cfg | grep CUR_COMMIT
Currently Committed          (CUR_COMMIT) = DISABLED
db2inst1:/dbhome/db2inst1$ db2 "select * from dept"
db2inst1:/dbhome/db2inst1$ db2 +c "update dept set DEPTNAME='OPERATIONS_test' where deptno='E11'"
DB20000I The SQL command completed successfully.
db2inst1:/dbhome/db2inst1$
```

Scenario脚本准备

✓ Terminate02

语句一直卡着等待没有执行

```
db2inst1:/dbhome/db2inst1$ db2 "select * from dept"

DEPTNO DEPTNAME                MGRNO ADMRDEPT LOCATION
-----
A00 SPIFFY COMPUTER SERVICE DIV. 000010 A00 -
B01 PLANNING                    000020 A00 -
C01 INFORMATION CENTER          000030 A00 -
D01 DEVELOPMENT CENTER          - A00 -
D11 MANUFACTURING SYSTEMS       000060 D01 -
D21 ADMINISTRATION SYSTEMS      000070 D01 -
E01 SUPPORT SERVICES            000050 A00 -
E11 OPERATIONS1                 000090 E01 -
E21 SOFTWARE SUPPORT            000100 E01 -
F22 BRANCH OFFICE F2            - E01 -
G22 BRANCH OFFICE G2            - E01 -
H22 BRANCH OFFICE H2            - E01 -
I22 BRANCH OFFICE I2            - E01 -
J22 BRANCH OFFICE J2            - E01 -

14 record(s) selected.

db2inst1:/dbhome/db2inst1$
db2inst1:/dbhome/db2inst1$ db2 "select * from dept"
```

Step1: 查看全体DB状况

- 全体DB状况的获取

- 输入小写“d”

```
[1]001:07:40,refresh=2secs(0.001) Database AIX,member=[1/1],DB2INST1:SAMPLE [qp=OFF]
[d=Y,a=N,e=N,p=ALL]
```

	25%	50%	75%	100%
MaxActSess	---			
SortMemory				
LogUsed	-			
FCM BuFlow				

```

Start Date Start Time Status Sthres Buffers FCMBuf OtherMem
2016/02/23 00:13:37 Active 0 10.5M 832.0K 140.8M

Sessions ActSess LockUsed LockEscals DeadLocks LogReads LogWrites
17 1 0% 0 0 0 0

L_Reads P_Reads HitRatio A_Reads Writes A_writes Lock wait
0 0 0.00% 0.00% 0 0 1

Sortheap Sortovf PctSortovf AvgPRdTime AvgDRdTime AvgPwrTime AvgDwrTime
0 0 0.00% 0.00 0.00 0.00 0.00

```

发生有1个Lock Wait

```
quit: q, Help: h g19aedrdb101i db2top 2.0
```

Step2: 查看session的状况

- 全体session状况的获取
 - 输入小写“l” 获取session的情况

[-]10:39:53,refresh=2secs(0.003) Sessions AIX,member=[1/1],DB2INST1:SAMPLE [d=Y,a=N,e=N,p=ALL] [qp=off]

②输入“l”

Application Handle(Stat)	Cpu% Total	IO% Total	Mem% Total	Application Status	Application Name	Delta RowsRead/s
1126(i)	0.00%	0.00%	7.14%	Now waiting in the application	db2bp	0
1132(l)	100.00%	0.00%	7.14%	Lock waiting	db2bp	0

可以确认当前时间点LockWait的session是否还存在

[-]10:42:29,refresh=2secs(0.001) Sessions AIX,member=[1/1],DB2INST1:SAMPLE [d=Y,a=N,e=N,p=ALL] [qp=off]

③输入“←”

Application Handle(Stat)	Sess Memory	Assoc. Agents	Paral. Degree	Lockwait (sec)	Locks Held	Sorts (sec)	Log Used	Delta RowsSelect/s	Fetch Count(Stmt)
1126(i)	320.0K	1	1	0	3	0	394	0	0
1132(l)	320.0K	1	1	735	4	0	0	0	0

查看LockWait的时间

Quit: q, Help: h

Step3: 查看lock情况

- 确认查看Lock的详细情况

- 输入大写"U"

```
[/]10:49:39,refresh=2secs(0.001) Locks AIX,member=[1/1],DB2INST1:SAMPLE [qp=off]
[d=Y,a=N,e=N,p=ALL]

Locks held.....:      6 [0.00]
Agents waiting...:      1
Appls Connected...:      2
```

Agent Id(State)	Application Name	Application Status
1132(I)	db2bp	Lock waiting
1126(I)	db2bp	UOW waiting in the application
1126(I)	db2bp	UOW waiting in the application
1126(I)	db2bp	UOW waiting in the application

④输入 "U"

ID1132的Agent 持有锁等待

```
[/]10:55:16,refresh=2secs(0.001) Locks AIX,member=[1/1],DB2INST1:SAMPLE [qp=off]
[d=Y,a=N,e=N,p=ALL]

Locks held.....:      6 [0.00]
Agents waiting...:      1
```

查看LockWait的整体情况

⑤输入 "←"

Agent Id(State)	Status	Object Name	Lock Mode	Object Type
1132(I)	Lock waiting	Internal Variation	S	Variation
1132(I)	Lock waiting	DB2INST1.DEPARTMENT[0]	NS [x]	Row
1132(I)	Lock waiting	DB2INST1.DEPARTMENT	IS	Table
1132(I)	Lock waiting	Internal Plan	S	Plan
1126(I)	UOW waiting in the application	DB2INST1.DEPARTMENT	X	Row
1126(I)	UOW waiting in the application	DB2INST1.DEPARTMENT	IX	Table
1126(I)	UOW waiting in the application	Internal Plan	S	Plan

quit: q, help: h

Step4: 查看lockchain的情况

- 确认锁链情况
 - 输入大写“L”

```
[ ] 10:58:52,refresh=2secs(0.000) Locks AIX,member=[1/1],DB2INST1:SAMPLE [qp=off]
[d=Y,a=N,e=N,p=ALL]
Blocker->Blocked Agent Chain
1126->1132
Press any key to resume...
Quit: q, Help: h Lock=10 (Entries=6), L: Lock Chain db2top 2.0
```

可以很容易的确认是那个 session 在锁等待中：
1132 在等待 1126 的完成

Step5: 查看lockwait的sql语句

- 确认Lockwait的具体SQL语句是什么
 - 输入小写“a”和AgentID查看

The screenshot shows a DB2 command line interface with the following content:

```
[*]11:08:35,refresh=2secs(0.001) Locks AIX,member=[1/1],DB2INST1:SAMPLI
[d=Y,a=N,e=N,p=ALL] [qp=OFF]
Please enter agent id: 1132
Locks held..... 6 [0.00]
Locks waiting... 1
Released..... 2
⑧输入AgentID
```

Agent Id(State)	Application Name	Application Status
1132(1)	db2bp	Lock waiting
1126(1)	db2bp	UOW waiting in the application
1126(1)	db2bp	UOW waiting in the application
1126(1)	db2bp	UOW waiting in the application

⑦输入“a”

```
[*]11:10:36,refresh=121!secs(0.001) Locks AIX,member=[1/1],DB2INST1:SAMPLI
[d=Y,a=N,e=N,p=ALL] [qp=OFF]
*LOCAL.db2inst1.160223012801, Lock waiting, blocked by 1126 on DB2INST1.DEP
select * from dept
Quit: Q.
```

⑧输入“e”可以获取此SQL的执行计划

输入“e”执行db2expln
 输入“x”执行db2exfmt
 输入“w”执行保存
 输入“E”编辑SQL问(用vi打开)
 输入“E”返回

db2top 历史信息收集和重放

- 一般情况，我们常用db2top实时检测数据库，但我们有些条件不可能一直盯着屏幕，这是可以考虑在晚上收集信息，第二天进行分析，类似回放录像。重新播放时候，我们可以直接跳到某个给定时间戳。
 - 通过大写"-C"，可以把db2top信息保存到文件，但之后需要交互输入 N/y ，所以不能后台进行，只有等待指定时间结束或者是<CTRL+C> 终止。

```
db2inst1:/dbhome/db2inst1/testdir$ db2top -d sample -C -m 5 -i 15
[11:38:28] Starting DB2 snapshot data collector, collection every 15 second(s), max duration 5 minute(s), max file growth/hour
100.0M, hit <CTRL+C> to cancel...
[11:38:28] Writing to 'db2snap-sample-AIX64.bin', should I create a named pipe instead of a file [N/y]?
[11:38:31] Creating 'db2snap-sample-AIX64.bin' as a normal file

[11:42:46] 1.0M written, time 258.165, 14.7M/hour
[11:43:31] Max duration reached, 1.2M bytes, time was 303.181...
[11:43:31] Snapshot data collection stored in 'db2snap-sample-AIX64.bin'
Exiting...
db2inst1:/dbhome/db2inst1/testdir$
db2inst1:/dbhome/db2inst1/testdir$ ls -ltr db2snap-sample-AIX64.bin
-rw-----  1 db2inst1 db2inst1  1308329 Feb 24 11:43 db2snap-sample-AIX64.bin
db2inst1:/dbhome/db2inst1/testdir$ db2top -d sample -f db2snap-sample-AIX64.bin
```

不加f的话默认收集的文件如下：<db2snap-<dbname>-<Machine><bits><.bin>

- 指定时间 db2top -d sample -f db2snap-sample-AIX64.bin /11:40:00

db2toprc 配置文件

- db2toprc 配置文件是用户生成的文件，用于在初始化时为 db2top 监视实用程序设置参数。（ db2toprc 是隐藏文件带“.” ）
 - db2top 实用程序将使用用户定义的变量 \$db2topRC 搜索 .db2toprc 文件的位置。如果该变量尚未设置，那么 db2top 将首先在当前目录中搜索 .db2toprc 文件，然后再在 home 目录中搜索该文件。 .db2toprc 文件是用户生成的文件。

```
#
# db2top configuration file
# On unix, should be located in $HOME/.db2toprc
# File generated by db2top-2.0
#
node=                # [-n] nodename
database=tpcc        # [-d] databasename
user=                # [-u] database user
password=            # [-p] user password (crypted)
schema=              # [-V] default schema for explains
interval=2           # [-i] sampling interval
active=OFF           # [-a] display active sessions only (on/off)
reset=OFF            # [-R] Reset snapshot at startup (on/off)
delta=ON             # [-k] Toggle display of delta/cumulative values (on/off)
graphic=ON           # True if terminal supports semi graphical characters
colors=ON            # True if terminal supports colors
gauge=ON             # display gauges (on/off)
port=8810            # Port for network collection
streamsize=100.0M   # Max collection size per hour (eg. 1024 or 1K : K, M
or G)
```

db2toprc 配置文件

- db2toprc 配置文件是用户生成的文件，用于在初始化时为 db2top 监视实用程序设置参数。（ db2toprc 是隐藏文件带“.” ）
 - db2top 实用程序将使用用户定义的变量 \$db2topRC 搜索 .db2toprc 文件的位置。如果该变量尚未设置，那么 db2top 将首先在当前目录中搜索 .db2toprc 文件，然后再在 home 目录中搜索该文件。 .db2toprc 文件是用户生成的文件。（运行db2top后输入“ w” 可以保存生成当前.db2toprc文件）

```

db2inst1:/dbhome/db2inst1$ ls -ltr .db2toprc
-rw-----  1 db2inst1 db2inst1    1657 Feb 24 11:34 .db2toprc
db2inst1:/dbhome/db2inst1$ cat .db2toprc
#
# db2top configuration file
# On unix, should be located in $HOME/.db2toprc
# File generated by db2top-2.0
#
node=                # [-n] nodename
database=sample     # [-d] databasename
user=                # [-u] database user
password=            # [-p] user password (crypted)
schema=              # [-V] default schema for explains
interval=2          # [-i] sampling interval
active=OFF           # [-a] display active sessions only (on/off)
reset=OFF            # [-R] Reset snapshot at startup (on/off)
delta=ON             # [-k] Toggle display of delta/cumulative values (on/off)
graphic=ON           # True if terminal supports semi graphical characters
colors=ON            # True if terminal supports colors
gauge=ON             # display gauges (on/off)
port=8810            # Port for network collection
streamsize=100.0M   # Max collection size per hour (eg. 1024 or 1K : K, M or G)

# Ordering of information in sessions screen
sessions=sort=1a
# Ordering of information in tables screen
tables=sort=7a
# Ordering of information in tablespaces screen
tablespaces=sort=34a
# Ordering of information in bufferpools screen
bufferpools=sort=17a

```

db2toprc 配置文件

- 样本 .db2toprc 文件

```
node= # [-n] 节点名
database=sample # [-d] 数据库名称
user= # [-u] 数据库用户
password= # [-p] 用户密码 (加密)
schema= # [-V] 说明的缺省模式
interval=2 # [-i] 采样时间间隔
active=OFF # [-a] 仅显示活动会话 (打开/关闭)
reset=OFF # [-R] 在启动时重置快照 (打开/关闭)
delta=ON # [-k] 切换增量值/累积值的显示 (打开/关闭)
gauge=ON # 在会话列表上显示图表 (打开/关闭)
colors=ON # 如果终端支持色彩, 那么为 True。如果它可以用色彩显示信息, 那么通知 GE_WRS
graphic=ON # 如果终端支持半图解字符, 那么为 True (打开/关闭)。
port= # 用于网络收集的端口
streamsize=size # 每小时的最大收集大小 (例如, 1024 或 1K : K、M 或 G)
```

注意点：

- ✓ 由于db2top所捕捉历史信息只能由db2top解析，不能直接转换为用户可以阅读的文本文件，在普调性能收集中建议使用普通的快照和db2pd
- ✓ 在多分区数据库系统中，需要谨慎使用db2top。因为db2top每次都是刷新实例级别的快照，如果有几百个分区的数据库做全局快照需要很大的内存开销，造成系统性能问题。所以需要指定某一个分区使用db2top
- ✓ db2top左上角d=Y表示当前要显示差值，如果想要显示绝对值，输入小写“k”
- ✓ 模拟Lockwait也可以使用下面示例：

Terninate01 :

```
db2 +c "create table t1(c1 int) "  
db2 +c "insert into t1 values(1)"
```

Terninate02 :

```
db2 +c "select * from t1 with RR "
```

未完待续



<http://3p.blogspot.com/> <http://yabilet.com/> <http://Dw3L3y.com/> <http://4444444400.com/> <http://0000000000.com/>

