



IBM Chat with Lab for Greater China Group

- **Host:** Frank Ning, Manager, DB2 LUW Install and Up/Running Development
- **Executive introduction (audio)**
 - Sal Vella, Vice President, Development, Distributed Data Servers and Data Warehousing
- **Presentation: Problem Determination and Database monitoring using db2pd**
 - Amy Tang, Jun Tang, Software developers, DB2 Reliability, Availability, Serviceability and Problem Determination (RAS/PD) team

Executive Introduction



Sal Vella

Vice President, Development, Distributed Data Servers
and Data Warehousing

IBM Software Group



Problem determination and database monitoring using db2pd tool

- Chat with the Toronto Lab for the Greater China Group

Amy Tang and Jun Tang
Software developers, DB2 Reliability, Availability, Serviceability and
Problem Determination (RAS/PD) team

December 8, 2010

© 2009 IBM Corporation



Agenda

- **RASPD Team and mission**
- **DB2 problem determination and monitoring by db2pd tool**
- **Examples: DB2 troubleshoot with db2pd tool**
- **DB2 Troubleshooting and Problem Determination Resources**
- **What's new in db2pd for pureScale (V98)**

RASPD team and our mission

- **RASPD**
 - **Reliability, Availability, Serviceability and Problem Determination**
 - **RAS/PD affects every aspect of our business**

- **Our mission:**
 - **reduce problem determination cost for DB2**
 - **Diagnostic APIs (trace, log ...)**
 - **Diagnostic tools for both DB2 developer and customer**
 - **db2trc, db2pd, db2pdcfg, db2dart, inspect, db2top, db2cos, db2fodc, db2support, db2cklog ...**



Problem determination and database monitoring using db2pd tool

- What does db2pd represent
 - DB2 Problem Determination
- An easy, yet powerful DB2 monitor and troubleshooting tool
- Executed from the command line, monitoring different scenarios, gives the user a closer view into the DB2 engine
- db2pd is a “read-only”, non-intrusive tool
 - collects information without acquiring any latches or using any engine resources.
 - Runs very quickly and runs outside of the engines (even works on a hung engine).
- Used by both customers and DB2 support / developers to monitor and troubleshoot.



Summary of Usages

- Monitoring OS
- Determine if instance or database is up
- Display DBMCFG and DBCFG
- Monitoring memory usage
- Detect and monitoring potential memory leak
- Monitoring buffer pool
- List all EDUs for an instance/partition
- Monitoring utilities
- Monitoring recovery
- Display index statistics
- Detect full-table scan vs index scans
- Determine table access ratio
- Monitoring overflow records
- Monitoring Table Reorgs
- Monitoring log usage
- Monitoring Tablespaces and Containers
- Monitoring locks
- Monitoring progress and behavior of DB2 agents
- Monitoring progress and behavior of applications
- Detecting hangs for specific applications
- Monitoring transactions
- Monitoring application progress
- Monitoring currently executing dynamic SQL statements
- Verifying isolation level
- Collecting call stacks
- Monitor latch contention



Monitoring OS: db2pd -osinfo

Operating System Information:

OSName: Linux
 NodeName: hotel49
 Version: 2
 Release: 6
 Machine: x86_64

CPU Information:

TotalCPU	OnlineCPU	ConfigCPU	Speed(MHz)	HMTDegree	Cores/Socket
4	4	4	1998	1	2

Physical Memory and Swap (Megabytes):

TotalMem	FreeMem	AvailMem	TotalSwap	FreeSwap
24109	8043	n/a	20490	20252

Virtual Memory (Megabytes):

Total	Reserved	Available	Free
44599	n/a	n/a	28295

Message Queue Information:

MsgSeg	MsgMax	MsgMap	MsgMni	MsgTql	MsgMnb	MsgSsz
n/a	65536	65536	24576	65536	65536	16

Shared Memory Information:

ShmMax	ShmMin	ShmIds	ShmSeg
25279922176	1	6144	6144

Semaphore Information:

SemMap	SemMni	SemMns	SemMnu	SemMsl	SemOpm	SemUme	SemUsz	SemVmx	SemAem
256000	6144	256000	256000	250	32	n/a	20	32767	32767

CPU Load Information:

Short	Medium	Long
1.420000	1.180000	1.430000

CPU Usage Information (percent):

Total	Usr	Sys	Wait	Idle
35.500000	n/a	n/a	n/a	64.500000

Determine whether instance or database is up

- **"db2pd -" tells whether instance is up and for how long**

- **db2pd -**

```
Database Partition 0 -- Active -- Up 4 days 05:34:53 Date 04/25/2010 15:12:04
```

- **"db2pd -db <database> -" tells how long the database has been active**

- **db2pd -db sample -**

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 19:11:07 Date  
04/25/2010 15:12:26
```

- **db2pd -alldbs -**

```
Database Partition 0 -- Database XMLDB -- Active -- Up 0 days 19:11:25 Date  
04/25/2010 15:14:30
```

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 19:11:43 Date  
04/25/2010 15:14:30
```



Database Manager Configuration (-dbmcfg)

Database Partition 0 -- Active -- Up 0 days 00:09:38 -- Date 11/19/2010 13:07:12

```
Database Manager Configuration Settings:
Description                Memory Value                Disk Value
RELEASE                    0xd00                       0xd00
CPUSPEED(millisecond/instruction) 4.000000e-05                4.000000e-05
COMM_BANDWIDTH(MB/sec)    1.250000e+00                1.250000e+00
NUMDB                      8                            8
FEDERATED                  NO                            NO
TP_MON_NAME
DFT_ACCOUNT_STR
JDK_PATH (memory)         /home/hotel49/juntang/sqllib/java/jdk64
JDK_PATH (disk)           /home/hotel49/juntang/sqllib/java/jdk64
DIAGLEVEL                  3                            3
NOTIFYLEVEL                3                            3
DIAGPATH (memory)
DIAGPATH (disk)
DIAGSIZE (MB)             0                            0
DFT_MON_BUFPOOL           OFF                           OFF
DFT_MON_LOCK              OFF                           OFF
DFT_MON_SORT              OFF                           OFF
DFT_MON_STMT              OFF                           OFF
DFT_MON_TABLE             OFF                           OFF
DFT_MON_TIMESTAMP        ON                            ON
DFT_MON_UOW               OFF                           OFF
HEALTH_MON                ON                            ON
SYSADM_GROUP (memory)    PDXDB2
SYSADM_GROUP (disk)     PDXDB2
```



Database Configuration (-dbcfg -db dbName)

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:02:54 --
Date 11/19/2010 13:09:55

Database Configuration Settings:

Description	Memory Value	Disk Value
DB configuration release level	0xd00	0xd00
Database release level	0xd00	0xd00
Database territory	US	US
Database code page	1208	1208
Database code set	UTF-8	UTF-8
Database country/region code	1	1
Database collating sequence	IDENTITY	IDENTITY
ALT_COLLATE	NON_UNIQUE	NON_UNIQUE
NUMBER_COMPAT	OFF	OFF
VARCHAR2_COMPAT	OFF	OFF
DATE_COMPAT	OFF	OFF
Database page size	8192	8192
DYN_QUERY_MGMT	DISABLE	DISABLE
Statement concentrator	OFF	OFF
DISCOVER_DB	ENABLE	ENABLE
Restrict access	NO	NO
DFT_QUERYOPT	5	5
DFT_DEGREE	1	1
DFT_SQLMATHWARN	NO	NO
DFT_REFRESH_AGE	0	0



Monitoring memory usage

- **db2pd -memsets -mempools**
- **reports statistics about DB2 Memory Sets and Memory Pools which helps in understanding memory usage**

Memory Sets:

Name	Address	Id	Size(Kb)	Key	DBP
DBMS	0x0000000200000000	989069392	55424	0x6BADCB61	0
FMP	0x0000000210000000	989102150	22592	0x0	0
Trace	0x0000000000000000	989036622	33610	0x6BADCB74	0

Type	Unrsv(Kb)	Used(Kb)	HWM(Kb)	Cmt(Kb)	Uncmt(Kb)
0	6592	22208	22272	22272	33152
0	2	0	960	22592	0
-1	0	33610	0	33610	0

Memory Pools:

Address	MemSet	PoolName	Id	Overhead	LogSz	LogUpBnd
0x0000000200001408	DBMS	fcm	74	0	0	652502
0x00000002000012C0	DBMS	fcmsess	77	65344	1941024	2228224
0x0000000200001178	DBMS	fcmchan	79	65344	394752	643072
0x0000000200001030	DBMS	fcmbp	13	65344	788992	1056768

. . .

LogHWM	PhySz	PhyUpBnd	PhyHWM	Bnd	BlkCnt	CfgParm
0	0	655360	0	Ovf	0	n/a
1941024	2097152	2228224	2097152	Ovf	3	n/a
394752	524288	655360	524288	Ovf	3	n/a
788992	983040	1114112	983040	Ovf	3	n/a

. . .



Monitoring memory usage (cont)

■ db2pd –memblocks

- Reports all memory blocks in DBMS set (list)
- Followed by the sorted 'per-pool' output

Memory blocks sorted by size for ostrack pool:

PoolID	PoolName	TotalSize(Bytes)	TotalCount	LOC	File
57	ostrack	5160048	1	3047	698130716
57	ostrack	240048	1	3034	698130716
57	ostrack	240	1	2983	698130716
57	ostrack	80	1	2999	698130716
57	ostrack	80	1	2970	698130716
57	ostrack	80	1	3009	698130716

Total size for ostrack pool: 5400576 bytes

- Final section sorts the consumers of memory for the entire set

All memory consumers in DBMS memory set:

PoolID	PoolName	TotalSize(Bytes)	%Bytes	TotalCount	%Count	LOC	File
57	ostrack	5160048	71.90	1	0.07	3047	698130716
50	sqlch	778496	10.85	1	0.07	202	2576467555
50	sqlch	271784	3.79	1	0.07	260	2576467555
57	ostrack	240048	3.34	1	0.07	3034	698130716
50	sqlch	144464	2.01	1	0.07	217	2576467555
69	krcbh	73640	1.03	5	0.36	547	4210081592



Detect and monitoring potential memory leak

- Run “db2pd -memblocks” to get a snapshot of allocated memory
- Look for TotalSize and TotalCount columns

Top memory consumers in Database memory set:

Memory blocks sorted by size:

PoolID	PoolName	TotalSize	TotalCount	LOC	File
16	bph	1791897600	6299	1165	4210081592
16	bph	189219200	18	436	698130716
4	lockh	21481152	1	636	2576467555
2	dbh	14708440	1	1111	586211086
2	dbh	14247808	222622	658	698130716 ←-----
2	dbh	3787328	32	810	356241690
←Next Iteration→					
16	bph	1791897600	6299	1165	4210081592
16	bph	189219200	18	436	698130716
4	lockh	21481152	1	636	2576467555
2	dbh	14951360	233615	658	698130716 ←-----
2	dbh	14708440	1	1111	586211086
←Next Iteration→					
16	bph	1791897600	6299	1165	4210081592
16	bph	189219200	18	436	698130716
4	lockh	21481152	1	636	2576467555
2	dbh	18925568	295712	658	698130716 ←-----
2	dbh	14708440	1	1111	586211086



Monitoring buffer pool

- **Determine whether we are spending time flushing buffers**
 - due to space constraint or poor allocation of pools
 - It's needed to identify areas for tuning

➤ **db2pd -buffer -db sample**

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 03:20:43 -- Date
04/23/2010 23:16:25
```

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	...
0x00002B9AED5F25A0	1	IBMDEFAULTBP	8192	1000	0	

BlkSize	NumTbsp	PgsToRemov	CurrentSz	PostAlter	SuspndTSCT	Automatic
0	7	0	1000	1000	0	False

```
. . .
. . .
```



Monitoring buffer pool (cont)

• • •
• • •

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

BPID	DatLRds	DatPRds	HitRatio	TmpDatLRds	TmpDatPRds	HitRatio	IdxLRds	IdxPRds	HitRatio	TmpIdxLRds	TmpIdxPRds
1	78	22	71.79%	0	0	00.00%	100	58	42.00%	0	0
	HitRatio										
	00.00%										
BPID	DataWrts	IdxWrts	DirRds	DirRdReqs	DirRdTime	DirWrts	DirWrtReqs	DirWrtTime			
1	0	0	42	5	0	0	0	0			
BPID	AsDatRds	AsDatRdReq	AsIdxRds	AsIdxRdReq	AsRdTime	AsDatWrts	AsIdxWrts	AsWrtTime			
1	0	0	0	0	0	0	0	0			
BPID	TotRdTime	TotWrtTime	VectIORds	VectIOReq	BlockIORds	BlockIOReq	PhyPgMaps	FilesClose	NoVictAvl		
1	104	0	0	0	0	0	0	8	0	0	0
	UnRdPFetch										

- DatLRds (DatPRds) → number of logical (physical) data page reads for this bufferpool
- Hit ratio for data pages given the above logical and physical reads
- Same for index pages: IdxLRds, IdxPRds, HitRatio

Monitoring buffer pool (cont)

- Filtering bufferpools output by bufferpool ID

```
-bufferpools <bpID>
```

- db2pd -db sample -bufferpools 4099

```
Address          Id  Name          PageSz    PA-NumPgs  BA-NumPgs  BlkSize  . . .
0x07800000203C99A0 4099 IBMSYSTEMBP32K 32768      16          0         0

  NumTbsp    PgsToRemov  CurrentSz  PostAlter  SuspndTSCt
    0         0           16         16         0
```

Bufferpool Statistics for bufferpool 4099 (when BUFFERPOOL monitor switch is ON):



db2pd -pages

- db2pd -db sample -pages
 - Pages for all bufferpools

- db2pd -db sample -pages [<bpID>]

- Monitor bufferpool behavior
 - Tells which pages are in the bufferpool
 - Use to determine what is in the bufferpool that is the cause for the hit ratio to be lower than you expect

- Allows user to check how many pages each object (table, index, etc) has within any particular bufferpool

- might help to detect a problem, such as an insufficient number of buffers in the buffer pool or high “read aheads”

db2pd -pages: example

- db2pd -db sample -page (db2pd -db sample -page summary, for summary only)

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:01:01

Bufferpool Pages:

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

Pages for all bufferpools:

Address	BPID	TbpaceID	TbpacePgNum	ObjID	ObjPgNum	ObjClass	ObjType	Dirty	Prefetched
0x07800000204DD040	1	0	6	19	6	Perm	Index	N	N
0x07800000204DD0F0	1	0	0	19	0	Perm	LOBA	N	N
0x07800000204DD1A0	1	0	0	19	0	Perm	Index	N	N
0x07800000204DD250	1	0	1	19	1	Perm	Index	N	N
0x07800000204DD300	1	0	2	19	2	Perm	Index	N	N
0x07800000204DD3B0	1	0	0	1	0	Perm	Data	N	N
0x07800000204DD460	1	0	4	19	4	Perm	Index	N	N
0x07800000204DD510	1	0	0	19	0	Perm	Data	N	N
0x07800000204DDB40	1	0	1	87	1	Perm	Index	N	N
0x07800000204DDBF0	1	0	7	19	7	Perm	Index	N	N
0x07800000204DDCA0	1	0	8	19	8	Perm	Index	N	N
0x07800000204DDD50	1	0	9	19	9	Perm	Index	N	N
0x07800000204DDE00	1	0	10	19	10	Perm	Index	N	N

<snip>

Total number of pages: 80

Summary info for all bufferpools:

BPID	TbpaceID	ObjID	Total	Dirty	Permanent	Temporary	Data	Index	LongField	XMLData	SMP	LOB	LOBA	BMP
1	0	1	2	0	1	0	2	0	0	0	0	0	0	0
1	0	5	51	8	47	0	30	18	0	0	0	1	2	0
1	0	6	14	0	12	0	4	10	0	0	0	0	0	0
1	0	7	26	12	23	0	13	11	0	0	0	0	2	0

<snip>



List all EDUs for an instance/partition : db2pd -edus

Database Partition 0 -- Active -- Up 0 days 00:47:59 -- Date 11/19/2010 13:45:33

List of all EDUs for database partition 0

db2sysc PID: 19699

db2wdog PID: 19688

db2acd PID: 19754

EDU ID	TID	Kernel TID	EDU Name	USR (s)	SYS (s)
141	46957263317312	8810	db2agnta (SAMPLE) 0	0.000000	0.000000
140	46957296871744	28012	db2pfchr (SAMPLE) 0	0.000000	0.000000
139	46957313648960	28011	db2pfchr (SAMPLE) 0	0.000000	0.000000
138	46957309454656	28010	db2pfchr (SAMPLE) 0	0.000000	0.000000
137	46957301066048	28009	db2pclnr (SAMPLE) 0	0.000000	0.000000
136	46957305260352	28008	db2dlock (SAMPLE) 0	0.000000	0.000000
135	46957317843264	28007	db2glock (SAMPLE) 0	0.000000	0.000000
134	46957322037568	28006	db2lfr (SAMPLE) 0	0.000000	0.000000
133	46957330426176	28005	db2loggw (SAMPLE) 0	0.000000	0.000000
132	46957326231872	28004	db2loggr (SAMPLE) 0	0.000000	0.000000
131	46957233957184	22448	db2lused (SAMPLE) 0	0.000000	0.000000
126	46957238151488	22443	db2agntdp (SAMPLE) 0	0.000000	0.000000
121	46957242345792	22422	db2taskd (SAMPLE) 0	0.000000	0.000000
120	46957246540096	22421	db2evmgi (DB2DETAILDEADLOCK) 0	0.000000	0.000000
119	46957250734400	22420	db2agntdp (SAMPLE) 0	0.000000	0.000000
118	46957254928704	22419	db2agntdp (SAMPLE) 0	0.000000	0.000000
117	46957259123008	22418	db2fw0 (SAMPLE) 0	0.000000	0.000000
93	46957267511616	22378	db2agntdp (SAMPLE) 0	0.000000	0.000000
92	46957271705920	22377	db2wlmd (SAMPLE) 0	0.000000	0.000000
45	46957338814784	19759	db2agent (SAMPLE) 0	0.000000	0.000000
44	46957343009088	19752	db2resync 0	0.000000	0.000000
43	46957347203392	19748	db2ipccm 0	0.000000	0.000000
.					

Monitoring utilities: db2pd -utilities

Utilities:

Address	ID	Type	State	Invoker	Priority
0x000000020121EA00	1	BACKUP	0	0	0

StartTime	DBName	NumPhases	CurPhase	Description
Mon Apr 26 10:44:09	SAMPLE	1	1	offline db

Progress:

Address	ID	PhaseNum	CompletedWork	TotalWork
0x000000020121F060	1	1	84262121 bytes	155421441 bytes

StartTime	Description
Mon Apr 26 10:44:09	n/a

- Utility types: BACKUP, RUNSTATS, REORG, RESTORE, CRASH_RECOVERY, ROLLFORWARD_RECOVERY, LOAD, RESTART_RECREATE_INDEX
 - db2 backup db sample
 - db2 restore db sample

- Use to monitor utilities' progress
 - Determine whether to throttle or unthrottle BACKUP or RUNSTATS utility (using SET UTIL_IMPACT_PRIORITY)

Recovery: db2pd -recovery

Recovery:

```

Recovery Status      0x00000401
Current Log          S0000005.LOG
Current LSN          000002551BEA
Job Type              ROLLFORWARD RECOVERY
Job ID                7
Job Start Time       (1107380474) Wed Feb  2 16:41:14 2005
Job Description       Database Rollforward Recovery
Invoker Type          User
Total Phases          2
Current Phase         1
  
```

Progress:

Address	PhaseNum	Description	StartTime	CompletedWork	TotalWork
0x0000000200667160	1	Forward	Wed Feb 2 16:41:14 2005	2268098 bytes	Unknown
0x0000000200667258	2	Backward	NotStarted	0 bytes	Unknown

Monitoring recovery

db2pd –recovery shows several counters to make sure recovery is progressing:

- Current Log and Current LSN provide the log position
- CompletedWork counts the number of bytes completed thus far



Index Statistics: db2pd -tcbstats all

- Useful for performance tuning
- Numerous statistics are reported to provide characteristics about each index's use
 - Only database activation/deactivation will reset these statistics
- “db2pd –tcbstats all” or “db2pd –tcbstats index”

TCB Index Stats:

Address	TableName	IID	PartID	EmpPgDel	RootSplits	BndrySplts	PseuEmptPg	EmPgMkdUsd
0x00002B38048A2D50	SYSTABLES	9	n/a	0	0	0	0	0
0x00002B38048A2D50	SYSTABLES	8	n/a	0	0	0	0	0
0x00002B38048A2D50	SYSTABLES	7	n/a	0	0	0	0	0

Scans	IxOnlyScns	KeyUpdates	InclUpdats	NonBndSpts	PgAllocs	Merges	PseuDels	DelClean	IntNodSpl
24	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0

- Scans
 - The number of scans against the index

Detect full-table scan vs index scans

- **Detect full-table scans for every table**

- **'db2pd -tcb -db <dbname>'**

- db2 "select * from employee"
- check the "Scans" column in "TCB Table States :" section
- db2pd -tcb -db sample | awk '/TCB Table Stats/ { found =1 } found==1 { print}' | grep -i employee | awk '{print "Scans: ", \$4}'
- Scans: 1

- **Detect number of index scans**

- **'TCB Index Stats' portion of the 'db2pd -tcb index -db <dbname>' output**

- db2 "CREATE INDEX LNAME ON EMPLOYEE (LASTNAME ASC)"
- db2 "select * from employee order by lastname"
- Check the "Scans" column in "TCB Index States:" section
- db2pd -tcb index -db sample | awk '/TCB Index Stats/ { found =1 } found==1 { print}' | grep -i employee | awk '\$10 > 0 {print "Index ID:",\$3, "Scans: ", \$10}'
- Index ID: 3 Scans: 1

DB2 table access ratio

- **db2pd can help determine the access frequency of each table**
 - Operations such as select, update, insert and delete
 - Reported by `db2pd -db <dbname> -tcbstats`

- **Identify tables with most inserts done to them**
 - “Inserts” column in “TCB Table Stats” section
 - `db2pd -db sample -tcbstats | awk '/TCB Table Stats/ { found =1} found==1 { print}' | awk '/^0x/ { print $11, $2}' | sort -rn | head -5`
 - 36 STAFF3
 - 32 EMPLOYEE
 - 20 PROJECT
 - 7 SYSCOLUMNS
 - 1 SYSUSERAUTH



Monitoring overflow records

- **db2pd –tcbstats**
 - number of overflow records read: `OvFlReads`
 - number of overflow records created: `OvFlCrtes`
 - `db2pd -tcbstats -alldbs | egrep 'Address|^0x' | awk '{print $14, $15}'`

- **Number of overflow records read is important**
 - Number of overflow records that **exist** in a table can be reported by statistics on a table (`runstats`)
 - Overflow record is of interest if it's **read** very often (indicates data fragmentation)

- **db2pd –tcbstats reports**
 - if overflow records are read and
 - how frequently they are being created

- **If overflow records are often read, one may improve table performance by reorganizing the table using the REORG utility**



Monitoring Table Reorgs: db2pd -reorgs

- Table reorg output including tablespace id, table id, table name, phases, counters, type (offline/online), start time, and end time are reported

Table Reorg Information:

Address	TbSpaceID	TableID	PartID	MasterTbs	MasterTab
0x00002B20555DEA28	2	6	n/a	n/a	n/a

TableName	Type	IndexID	TempSpaceID
EMPLOYEE	Offline	0	2

Table Reorg Stats:

Address	TableName	Start	End
0x00002B20555DEA28	EMPLOYEE	04/26/2010 10:52:22	04/26/2010 10:52:22

PhaseStart	MaxPhase	Phase	CurCount	MaxCount	Status	Completion
04/26/2010 10:52:22	3	IdxRecreat	0	0	Done	0

- Phase field (only applies to offline table reorganization)
 - The phase of the table reorganization: **Sort, Build, Replace, InxRecreat**
- Status field (only applies to online table reorganization)
 - status of an online table reorganization: **Started, Paused, Stopped, Done, Truncat**
 - "Done" status indicates that the reorg utility has been completed
- Completion field
 - success indicator for the table reorganization. Possible values:
 - 0. The table reorganization completed successfully
 - 1. The table reorganization failed



Monitoring log usage

```
$ db2pd -logs -db <dbname>
Logs:
Current Log Number          0
Pages Written                50
Cur Commit Disk Log Reads  0
Cur Commit Total Log Reads 0
Method 1 Archive Status     n/a
Method 1 Next Log to Archive n/a
Method 1 First Failure      n/a
Method 2 Archive Status     n/a
Method 2 Next Log to Archive n/a
Method 2 First Failure      n/a
Log Chain ID                0
Current LSN                  0x0000000001F72ABF
```

Address	StartLSN	State	Size	Pages	Filename
0x00002B38017E62D0	0000000001F40010	0x00000000	1000	1000	S0000000.LOG
0x00002B38017FCD70	0000000002328010	0x00000000	1000	1000	S0000001.LOG
0x00002B38017FD5D0	0000000002710010	0x00000000	1000	1000	S0000002.LOG

- **Two problems can be identified with this output**
 - Problem with archiving
 - if Archive Status is set to Failure, the most recent log archive failed
 - If First Failure is set, ongoing archive failure is preventing logs from archiving
 - Log archiving is proceeding very slowly
 - Next Log to Archive will be behind Current Log Number (this can cause the log path to fill up completely)
 - Monitor 'Next Log to Archive' compared to 'Current Log Number'
 - If next log is 3 and current is 5, then logs 3 and 4 haven't been archived yet
 - Log 5 is the current log being written into



Monitoring Tablespaces and Containers

- **Single tablespace can be monitored**

- `db2pd -tab[lespaces] <tablespaceID> -rep[eat] <numSecs>`
- `db2 "create tablespace dms1 managed by database using (file 'tbspace1' 1M)"`
- `db2pd -db sample -tab | grep DMS1 | awk '{print $2, $15}'`
`6 DMS1 ← tablespace ID is 6`
- `db2pd -db sample | perl -ane 'if (/TotalPgs * UsablePgs * UsedPgs/ .. /^$/) { print "$F[3] $F[4]\n"}'`
`UsablePgs UsedPgs`
`224 96`
- `db2 "insert into staff3 select * from staff"`
`...repeat ...`
- `db2pd -db sample -tab 6 | perl -ane 'if (/TotalPgs * UsablePgs * UsedPgs/ .. /^$/) { print "$F[3] $F[4]\n" }'`
`UsablePgs UsedPgs`
`224 160`



Monitoring locks

- Lockname <==> hex representation of the physical object that is being waited on
- `db2pd -db sample -lock showlocks`

```
Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:10:42
```

Locks:

```
... Lockname                               Type
... 000200030000001A00000000052 Row      TbspaceID 2 TableID 3 RecordID 0x1A
... 000200030000001F00000000052 Row      TbspaceID 2 TableID 3 RecordID 0x1F
... 53514C4332453036C8324ABC41 Internal P Pkg UniqueID 53514c43 32453036 Name c8324abc
... 53514C4332453036C8324ABC41 Internal P Pkg UniqueID 53514c43 32453036 Name c8324abc
... 0002000D0000000500000000052 Row      TbspaceID 2 TableID 13 RecordID 0x5
... 000200030000000000000000054 Table    TbspaceID 2 TableID 3
... 0002000D0000000000000000054 Table    TbspaceID 2 TableID 13
```

- **'showlocks' suboption will expand the lockname into meaningful explanations**

To determine who is holding a lock in your database

- `db2pd -database sample -locks -transactions -agents -file lock.txt`
- -agents will contain UserID for transaction handle that is holding a lock (status is G (granted))

- Map lock info to a table name
 - Use TableID from Lockname or showlocks output
 - `db2pd -tcbstats -db sample | awk '/Address/^0x/ { print $2,$3,$7}'`

```
TbspaceID TableID TableName
0          18          SYSROUTINES
0          81          SYSROUTINEPROPERTYI
2          4           DEPARTMENT
```

Monitoring progress and behavior of DB2 agents

- `db2pd -agents [[agent=<agentid>] | [app=<appHdl>]]`

Agents:

```
Current agents:      10
Idle agents:        0
Active coord agents: 10
Active agents total: 10
Pooled coord agents: 0
Pooled agents total: 0
```

Address	AppHandl	[nod-index]	AgentEDUID	Priority	Type	State
0x0000000201405DE0	7	[000-00007]	16	0	Coord	Inst-Active
0x0000000201470080	8	[000-00008]	27	0	Coord	Inst-Active
0x0000000201475DE0	9	[000-00009]	28	0	Coord	Inst-Active
0x00000002014A0080	10	[000-00010]	29	0	Coord	Inst-Active
0x00000002014A5DE0	11	[000-00011]	30	0	Coord	Inst-Active

ClientPid	Userid	ClientNm	Rowsread	Rowswrtn	LkTmOt	DBName
30359	juntang	db2bp	173	0	NotSet	RBTRAPDB
30359	juntang	db2stmm	0	0	NotSet	RBTRAPDB
30359	juntang	db2taskd	3	0	NotSet	RBTRAPDB
30359	juntang	db2wlmd	0	0	NotSet	RBTRAPDB
30359	juntang	db2lused	0	0	3	RBTRAPDB

Work with repeat Option: `db2pd -rep [num sec] [count]`

- **`db2pd -age -rep 10 3`**

Address	AppHandl	[nod-index]	AgentEUID	Priority	Type	State
0x0000000201405DE0	7	[000-00007]	16	0	Coord	Inst-Active
0x0000000201470080	8	[000-00008]	27	0	Coord	Inst-Active
0x0000000201475DE0	9	[000-00009]	28	0	Coord	Inst-Active

ClientPid	Userid	ClientNm	Rowsread	Rowswrtn	LkTmOt	DBName
13885	juntang	db2bp	219	15	NotSet	RBTRAPDB
13885	juntang	db2stmm	0	0	NotSet	RBTRAPDB
13885	juntang	db2taskd	3	0	NotSet	RBTRAPDB

- **Repeat option is handy for watching activities. Example above watches agent's reads and writes**
- **Combine -repeat option with the file redirection**
 - `db2pd -age file=agents.out -rep 10 3`
- **Combine multiple options and use mixed scope options**
 - `db2pd -db sample -loc -tra -age -fil lock.txt`
 - Use `-file` for multiple options

How to deal with too many db2agent processes (SQL1226N)

- **If many db2agent processes remain attached to an instance**
 - They may use up all of the agents (MAXAGENTS)
 - New connections will trigger SQL1226N
 - “The maximum number of client connections are already started”
 - ADM7009E error message will be logged in a notify log

- **Use “db2pd –agents” to find AppHandl, ClientPid, Userid and ClientNm**
 - **db2pd -agents | awk '/Address|^0x/ { print \$2, \$8, \$9, \$10}'**

AppHandl	ClientPid	Userid	ClientNm
643	3612794	dabrashk	db2bp
642	3227856	dabrashk	db2bp

- **Force the identified AppHandl off with command**
 - db2 "force application (642)"



Monitoring transactions: db2pd -transactions

Transactions:

Address	AppHandl	[nod-index]	TranHdl	Locks	State	Tflag
0x000000022026D980	797	[000-00797]	2	108	WRITE	0x00000000
0x000000022026E600	806	[000-00806]	3	157	WRITE	0x00000000
0x000000022026F280	807	[000-00807]	4	90	WRITE	0x00000000

Tflag2	Firstlsn	Lastlsn	LogSpace	SpaceReserved	...
0x00000000	0x000001072262	0x0000010B2C8C	4518	95450	
0x00000000	0x000001057574	0x0000010B3340	6576	139670	
0x00000000	0x00000107CF0C	0x0000010B2FDE	3762	79266	

Useful for determining the amount of resources a transaction is using

- db2pd -transactions provides
 - number of locks
 - first lsn, last lsn
 - Log Sequence Number represents relative byte address, within the database log, for the first byte of the log record
 - logspace used (in pages)
 - space reserved (in pages)
- Monitor the progress and behavior of any transaction

Monitoring progress and behavior of applications

- `db2pd -applications -db dbName`
 - Use to map application to a coordinator agent
 - Use to determine a status of application
 - Use to map application to the dynamic SQL statement



Detecting hangs for specific applications

- **Take a few application snapshots a minute apart to determine**
 - what the status of the application is (`db2pd -app: Status`) and
 - whether any work is being done (`db2pd -agent: RowsRead/Wrtn`)
 - It is useful to have turned on all of the monitor switches prior to a re-creatable or recurring problem scenario
- **Use `db2pd -app` to determine status of an application (Status column)**
 - `db2pd -app -db sample | awk '/Address|^0x/ { print $6 }'`
- **If status is UOW Waiting, the hang is not occurring at the DB2 server**
 - The client application should be investigated to find out what it is waiting for.
 - In DPF environment, this may indicate a problem with another partition
- **If status is Executing and counters like rows-read/written are increasing, it is likely a performance issue**
- **If status is Lock-wait than it is a locking/concurrency issue**
 - Exception is the case when the application being waited on is in UOW Executing and making no progress
- **If status is Executing yet no counters are increasing, then the agent or agents servicing the application may be in an abnormal state**

† More diagnostics is needed

Monitoring application progress

- **Identifying slow or hanging applications**

- **Monitor rows read and written for agents**

- `db2pd -agents | awk '/Address|^0x/ { print $2, $11, $12, $14;}'`

AppHandl	Rowsread	Rowswrtn	DBName
51	109	58	SAMPLE
44	46	0	SAMPLE
9	0	0	n/a
8	126	107	SAMPLE
0	0	0	SAMPLE

- Use AppHandl to determine the application to take action on
 - Use `db2pd –dynamic` to find SQL statement
 - Use `db2pd –static` to find package



Monitoring application progress (cont'd)

- **Monitor start time of statements in db2pd –activestatements**

- db2pd -activestat -db sample | awk '/Address/ { print \$2,\$6,\$7,\$12,\$13 } /^0x/ {print \$2,\$6,\$7,substr(\$0,135);}'

```
AppHandl AnchID StmtUID StartTime LastRefTime
76      44      1      Sat Jan 7 17:57:35 2006 Sat Jan 7 17:57:35 2006
```

- Use AnchID and StmtUID to identify SQL statement
- Use AppHandl to identify application

- **Find “non-committing” transactions**

- db2pd -trans -db sample | awk '/Address|^0x/ { print \$2,\$9,\$10;}'

```
AppHandl Firstlsn Lastlsn
76      0x0001801CF000 0x000202999600
44      0x000177000C00 0x000177000C00
```

- Use the first and last LSN (log sequence number) of the transaction
 - db2flsn executable can be used to identify the log file for a specific lsn.

- **Find which application used biggest number of locks**

- Use AppHandl and Locks fields of db2pd -transactions
- db2pd -trans -db sample | awk '/^0x/ { print \$5, \$2}' | sort -rn

```
7 76
4 74
2 44
```



Monitoring currently executing dynamic SQL statements

➤ Example: db2 “select * from dept”

➤ db2pd -db sample -app

Applications:

Address	AppHandl	[nod-index]	NumAgents	CoorEUID	Status
0x0000000201410080	27	[000-00027]	1	35	UOW-Waiting
0x0000000201507F00	33	[000-00033]	1	68	ConnectCompleted

C-AnchID	C-StmtUID	L-AnchID	L-StmtUID	Appid	
0	0	769	1	*LOCAL.juntang.100426034408	. . .
0	0	0	0	*LOCAL.DB2.100426034414	. . .

➤ db2pd -db sample -dyn

Dynamic SQL Statements:

Address	AnchID	StmtUID	NumEnv	NumVar	NumRef	NumExe	Text
0x00002B38048AED80	769	1	1	1	1	1	select * from dept

Dynamic SQL Environments:

Address	AnchID	StmtUID	EnvID	Iso	QOpt	Blk
0x00002B38048AEEC0	769	1	1	CS	5	B

Dynamic SQL Variations:

Address	AnchID	StmtUID	EnvID	VarID	NumRef	Typ	Lockname	Val	Insert
0x00002B38048AF220	769	1	1	1	1	6	01000000010000000100206056	Y	2010-04-
Time		Sect	Size						
25-23.44.40.154410		6000							

Verifying isolation level

- **Current isolation level of dynamic SQL statement**

- “how can I tell what isolation level is being used ?”
- db2pd -db sample -dynamic
 - ISO column in Dynamic SQL Environments section

Dynamic SQL Environments:

Address	AnchID	StmtUID	EnvID	Iso	QOpt	Blk
0x0780000020CEBC40	41	1	1	CS	5	B
0x0780000020BB5D80	42	2	1	CS	5	B
0x0780000020CE2FC0	235	1	1	RR	5	B

- db2pd -activestatements
 - EffISO column
 - 0=RR,1=CS,2=UR and 3=RS



Collecting call stacks

- **Used to determine what DB2 agent is doing**
- **Call stack collection**
 - Stack traces (trap files) collection is dissociated from binary dump files
- **Produce stack trace for all EDUs or chosen EDUID**
 - `db2pd -stack [all|<EDUID>]`
 - Produces trap file(s) in the DIAGPATH directory
- **Produce dump file and stack trace for all EDUs or chosen EDUID**
 - `db2pd -dump [all| EDUID>]`



Monitor latch contention: db2pd -latches

- Latch tracking is always-on
- db2pd -latches

Latches:

Address	Holder	Waiter	Filename	LOC
0x07800000203C40C8	16	0	sqlldpool.C	529
0x07800000203C4190	16	0	sqlbpool.C	2254
0x07800000203C5678	16	0	sqlbistorage.h	5169

LatchType	HoldCount
SQLO_LT_SQLB_CLNR_PAUSE_CB_preventSuspendLatch	1
SQLO_LT_SQLB_PTBL_pool_table_latch	1
SQLO_LT_SQLB_PTBL_ptfLatches	1

- Holder is agent EDU ID
- LatchType is a latch identifier
- To group latches by holders and waiters
 - db2pd -latches group

Latch Holders:

Address	Holder	Filename	LOC	LatchType	HoldCount
0x07800000204A8A48	16	sqlbilatch.C	1150	SQLO_LT_SQLB_POOL_CB_writeLatch	1

Latch Waiters:

Address	Waiter	Filename	LOC	LatchType
---------	--------	----------	-----	-----------

Examples: DB2 troubleshoot with db2pd tool

- Lock contention
- Figuring out which application is using up tablespace
- Implicit temporary table space



Example 1: Monitoring lock contention

- **Monitor for slowdowns**
 - Important to figure out “who is waiting for whom”
 - Identify the lock owner to consider action of releasing the lock

App1

Update mytab set c1=20 where c1=2

db2pd -wlocks -db sample

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:42:34 -- Date 04/23/2010 20:38:16

Locks being waited on :

AppHndl	[nod-index]	TranHdl	Lockname	Type	Mode	Conv	Sts	CoorEDU	AppName	AuthID	AppID
7	[000-00007]	2	03000500050000000000000052	Row	..X		G	16	db2bp	JUNTANG	*LOCAL.juntang.100423235541
63	[000-00063]	12	03000500050000000000000052	Row	..X		W	37	db2bp	JUNTANG	*LOCAL.juntang.100424002456

App2

.....
delete from mytab

- Look for “G” status for lock holder, “W” is waiter
- App1 (AppHdl 7, tranHdl 2) is lock owner
- App2 (AppHdl 63, tranHdl 12) is lock waiter



Monitoring lock contention (cont)

- Use db2pd –apinfo to get more detail information
 - What SQL statement these two applications are running
- App1 (AppHdl 7, tranHdl 2) is lock owner
 - db2pd –apinfo 7 –db sample
- App2 (AppHdl 63, tranHdl 12) is lock waiter
 - db2pd –apinfo 63 –db sample
- Note:
- To capture the past history of a unit of work (UOW) including the SQL statement text for the applications, activate a deadlock event monitor using the statement history clause.
- For example, use one of the following statements:
 - *create event monitor testit for deadlocks with details history write to table*
- The event monitor with statement history capability affects all applications and increases the monitor heap usage by the DB2 database manager.



Monitoring lock contention (cont)

- db2pd -apinfo 7 -db sample

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:42:41

Application :

```
Address : 0x0000000201410080
AppHandl [nod-index] : 7 [000-00007]
TranHdl : 2
Application PID : 6376
Application Node Name : hotel49
IP Address: n/a
Connection Start Time : (1272066942)Fri Apr 23 19:55:42 2010
Client User ID : juntang
System Auth ID : JUNTANG
Coordinator EDU ID : 16
Coordinator Partition : 0
Number of Agents : 1
Locks timeout value : NotSet
Locks Escalation : No
Workload ID : 1
Workload Occurrence ID : 1
Trusted Context : n/a
Connection Trust Type : non trusted
Role Inherited : n/a
Application Status : UOW-Waiting
Application Name : db2bp
Application ID : *LOCAL.juntang.100423235541
ClientUserID : n/a
ClientWrkstnName : n/a
ClientApplName : n/a
ClientAccntng : n/a
CollectActData: N
CollectActPartition: C
SectionActuals: N
```

Monitoring lock contention (cont)

List of inactive statements of current UOW :

```
UOW-ID :          24
Activity ID :      1
Package Schema :  NULLID
Package Name :    SQLC2H21
Package Version :
Section Number :  203
SQL Type :        Dynamic
Isolation :       CS
Statement Type :  DML, Insert/Update/Delete
Statement :       update mytab set c1 = 20 where c1=2
```



Monitoring lock contention (cont)

- db2pd -apinfo 63 -db sample

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 00:43:12

Application :

```
Address : 0x0000000201580080
AppHandl [nod-index] : 63 [000-00063]
TranHdl : 12
Application PID : 23481
Application Node Name : hotel49
IP Address: n/a
Connection Start Time : (1272068696)Fri Apr 23 20:24:56 2010
Client User ID : juntang
System Auth ID : JUNTANG
Coordinator EDU ID : 37
Coordinator Partition : 0
Number of Agents : 1
Locks timeout value : NotSet
Locks Escalation : No
Workload ID : 1
Workload Occurrence ID : 2
Trusted Context : n/a
Connection Trust Type : non trusted
Role Inherited : n/a
Application Status : Lock-wait
Application Name : db2bp
Application ID : *LOCAL.juntang.100424002456
ClientUserID : n/a
ClientWrkstnName : n/a
ClientApplName : n/a
ClientAccntng : n/a
CollectActData: N
CollectActPartition: C
SectionActuals: N
```


Monitoring lock contention (cont)

List of active statements :

```
*UOW-ID :          3
Activity ID :      1
Package Schema :  NULLID
Package Name :    SQLC2H21
Package Version :
Section Number :  203
SQL Type :        Dynamic
Isolation :       CS
Statement Type :  DML, Insert/Update/Delete
Statement :       delete from mytab
```

Example 2: Figuring out which application is using up your tablespace

- Identify number of Inserts for table (here, temp table TEMP1)
 - db2pd –tcbstats

TCB Table Information:

Address	TbSpaceID	TableID	PartID	MasterTbs	MasterTab	TableName
0x0780000020B62AB0	3	2	n/a	3	2	TEMP1

SchemaNm	ObjClass	DataSize	LfSize	LobSize	XMLSize
SESSION	Temp	966	0	0	0

TCB Table Stats:

Address	TableName	Scans	UDI	PgReorgs	NoChgUpdts
0x0780000020B62AB0	TEMP1	0	0	0	0

Reads	FscrUpdates	Inserts	Updates	Deletes	OvFlReads	OvFlCrtes
0	0	43968	0	0	0	0



Figuring out which application is using up your tablespace (cont)

Map to tablespace 3 in db2pd -tablespaces output:

Tablespace 3 Configuration:

Address	Type	Content	PageSz	ExtentSz	Auto	Prefetch	BufID
0x0780000020B1B5A0	DMS	UsrTmp	4096	32	Yes	32	1

BufIDDisk	FSC	NumCntrs	MaxStripe	LastConsecPg	Name
1	On	1	0	31	TEMPSPACE2

Tablespace 3 Statistics:

Address	TotalPgs	UsablePgs	UsedPgs	PndFreePgs
0x0780000020B1B5A0	5000	4960	1088	0

FreePgs	HWM	State	MinRecTime	NQuiescers
3872	1088	0x00000000	0	0

Tablespace 3 Autoresize Statistics:

Address	AS	AR	InitSize	IncSize	IIP	MaxSize	LastResize	LRF
0x0780000020B1B5A0	No	No	0	0	No	0	None	No

Containers:

Address	ContainNum	Type	TotalPgs	UseablePgs	StripeSet
0x0780000020B1DCC0	0	File	5000	4960	0

Container
/home/db2inst1/tempspace2a

The FreePgs column shows that space is filling up. As the free pages value decreases, there is less space available. Notice also that the value for FreePgs plus the value for UsedPgs equals the value of UsablePgs.



Figuring out which application is using up your tablespace (cont)

- **Identify the dynamic sql statement using a table called TEMP1**
 - db2pd -db sample -dyn

Dynamic Cache:

```

Current Memory Used          1022197
Total Heap Size              1271398
Cache Overflow Flag         0
Number of References         237
Number of Statement Inserts  32
Number of Statement Deletes  13
Number of Variation Inserts  21
Number of Statements         19
  
```

Dynamic SQL Statements:

Address	AnchID	StmtUID	NumEnv	NumVar	NumRef	NumExe
0x0000000220A08C40	78	1	2	2	3	2
0x0000000220A8D960	253	1	1	1	24	24

Text

```

declare global temporary table temp1 (c1 char(6)) not logged
insert into session.temp1 values('TEST')
  
```



Figuring out which application is using up your tablespace (cont)

- Map this to **-app** output to identify the application

- db2pd -app -db sample

Applications:

Address	AppHandl	[nod-index]	NumAgents	CoorEDUID	Status
0x0000000200661840	501	[000-00501]	1	16	UOW-Waiting

C-AnchID	C-StmtUID	L-AnchID	L-StmtUID	Appid
0	0	253	1	*LOCAL.db2inst1.050202160426

- db2pd -agent output will show the number of rows written as verification

Address	AppHandl	[nod-index]	AgentEDUID	Priority	Type
0x0000000200698080	501	[000-00501]	16	0	Coord

State	ClientPid	Userid	ClientNm	Rowsread	Rowswrtn	LkTmOt	DBName
Inst-Active	26377	db2inst1	db2bp	22	45365	NotSet	SAMPLE

Example 3: Monitoring implicit temporary table space

- **Steps are different for the implicit temporary table**
- **Use db2pd -tcbstats to identify tables with large numbers of inserts**

TCB Table Information:

Address	TbSpaceID	TableID	PartID	MasterTbs	MasterTab	TableName	SchemaNm	ObjClass	DataSize
0x0780000020CC0D30	1	2	n/a	1	2	TEMP (00001,00002)	<30>	<JMC Temp	2470
0x0780000020CC14B0	1	3	n/a	1	3	TEMP (00001,00003)	<31>	<JMC Temp	2367
0x0780000020CC21B0	1	4	n/a	1	4	TEMP (00001,00004)	<30>	<JMC Temp	1872

TCB Table Stats:

Address	TableName	Scans	UDI	PgReorgs	NoChgUpdts	Reads	FscrUpdates	Inserts	...
0x0780000020CC0D30	TEMP (00001,00002)	0	0	0	0	0	0	43219	...
0x0780000020CC14B0	TEMP (00001,00003)	0	0	0	0	0	0	42485	...
0x0780000020CC21B0	TEMP (00001,00004)	0	0	0	0	0	0	0	...

- **Notice large number of inserts for implicit temporary tables**
 - tables with the naming convention "TEMP (TbSpaceID, TableID)"
 - Identify the application doing the work
 - values in the SchemaNm column have a naming convention of <AppHandl><SchemaNm>



Monitoring implicit temporary table space (cont)

- Map that info to the used space for table space 1
 - Use **db2pd –tablespaces**
 - Notice the **UsedPgs vs the UsablePgs** in the table space statistics

Tablespace Configuration:

Address	Id	Type	Content	PageSz	ExtentSz	Auto	Prefetch
0x07800000203FB5A0	1	SMS	SysTmp	4096	32	Yes	320
BufID	BufIDDisk	FSC	NumCntrs	MaxStripe	LastConsecPg	Name	
1	1	On	10	0	31	TEMPSPACE1	

Tablespace Statistics:

Address	Id	TotalPgs	UsablePgs	UsedPgs	PndFreePgs
0x07800000203FB5A0	1	6516	6516	6516	0
FreePgs	HWM	State	MinRecTime	NQuiescers	
0	0	0x00000000	0	0	

Tablespace Autoresize Statistics:

Address	Id	AS	AR	InitSize	IncSize	IIP	MaxSize	LastResize	LRF
0x07800000203FB5A0	1	No	No	0	0	No	0	None	No

Containers:

...

Monitoring implicit temporary table space (cont)

- Identify the application handles 30 and 31
 - Seen in the **-tcbstats** output
 - **db2pd -app**

Applications:

Address	AppHandle	[nod-index]	NumAgents	CoorEUID	Status
0x07800000006FB880	31	[000-00031]	1	18	UOW-Waiting
0x07800000006F9CE0	30	[000-00030]	1	27	UOW-Executing
C-AnchID	C-StmtUID	L-AnchID	L-StmtUID	Appid	
0	0	107	1	*LOCAL.db2inst1.051215214142	
107	1	107	1	*LOCAL.db2inst1.051215214013	

- Map this to the Dynamic SQL using **db2pd -dyn**

Dynamic SQL Statements:

Address	AnchID	StmtUID	NumEnv	NumVar	NumRef	NumExe
0x07800000020B296C0	107	1	1	1	43	43
Text						
select c1, c2 from test group by c1,c2						

DB2 Troubleshooting and Problem Determination Resources

- **db2pd -- Monitor and troubleshoot DB2 database command**
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011729.html>
- **Monitoring and Troubleshooting using db2pd tool**
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.trb.doc/doc/c0054595.html>
- **Troubleshooting and Support**
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.admin.trb.doc/doc/c0054273.html>
- **What's new for V9.7: Troubleshooting and problem determination enhancements**
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.db2.luw.wn.doc/doc/c0023244.html>
- **DB2 Product Support site**
 - <http://www-306.ibm.com/software/data/db2/udb/support/index.html>
- **DB2 APARs (Authorized Program Analysis Reports)**
 - <http://www-306.ibm.com/support/docview.wss?rs=71&uid=swg21412439>

What's New in db2pd for pureScale (V98)

- *cfpool*
- *cfinfo*
- *totalmem*
- transactions
- recovery
- logs
- utilities
- *member*
- *allmembers*

V98 DB2 Information Center:

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/index.jsp?topic=/com.ibm.db2.luw.admin.cmd.doc/doc/r0011729.html>

db2pd -cfinfo

```
$ db2pd -db sample -cfinfo lock primary
CF Server (128) Information
  Hostname                               = coralpib23.torolab.ibm.com
  Management Port                         = 19263
  Role                                    = Primary
CF Host Information
  Virtual Memory Used                     = 59784773632
  Virtual Memory Available                 = 30409539584
  Real Memory Used                         = 59611738112
  Real Memory Available                   = 26287607808
  Swap File Used                           = 173035520
  Swap File Available                       = 4121931776
  CPU Usage                               = 0
Total Space (4KB)                         = 128000
Free Space (4KB)                           = 1024
Frame Size (4KB)                           = 256
Configured Size (4KB)                       = 126976

Lock Structure Information
  Structure Name                           = db2.amytang.SAMPLE.lock
  SID                                       = 240
  Current Size (4KB)                       = 19200
  Target Size (4KB)                         = 19200
  Configured Size (4KB)                     = 19200
```



db2pd -totalmem

\$ db2pd -totalmem

Total Memory Statistics in KB

	Controller Automatic	Memory Limit	Current Usage	HWM Usage	Cached Memory
Member 0	Yes	71907892	1099840	1099840	311616
Restart Light Memory	Yes	2996160	2842560	2842560	39360
Total current usage:		3942400			
Total cached memory:		350976			

Additional Information

- **Feedback**
 - Presentation format and contents
 - Additional DB2 topics you are interested
 - Follow on questions for the presentation

- **Contact: fning@ca.ibm.com**

Questions?

Thank you !