

Exploring the Sysmaster - my new scripts

Lester Knutsen

Advanced DataTools Corporation lester@advancedatatools.com





Lester Knutsen



Lester Knutsen is President of Advanced DataTools Corporation, and has been building large data warehouse and business systems using Informix Database software since 1983. Lester focuses on large database performance tuning, training, and consulting. Lester is a member of the IBM Gold Consultant program and was presented with one of the Inaugural IBM Information Champion awards by IBM. Lester was one of the founders of the International Informix Users Group and the Washington Area Informix User Group.

> lester@advancedatatools.com www.advancedatatools.com 703-256-0267 x102

Agenda

The session will continue the series of presentations I have given over the past 20 years and focus on my latest and greatest scripts to monitor and tune your Informix Database Server using data in the Sysmaster database. We will also explore the systables and the sysadmin database in these scripts. Some of the scripts we will look at are:

- What indexes are used and not used?
- What is the performance of your Dbspaces?
- What is the performance of your Logical Logs?
- How do you estimate your Logical Log usages for High Availability?
- How do you monitor Update Statistics and when were they last run?
- What is the performance of your tables? Most used? Least used? Most Sequential Scans?
- What are the most expensive SQL queries running now?
- How large will my level 1 and 2 backups be?

Agenda – 8 New/Updated Sysmaster Scripts

- 1. Dbspace and Chunk I/O and Info
- 2. Checkpoint I/O and Status
- 3. Logical Logs Usage and Turnover
- 4. Server Performance Ratios since restart
- 5. Update Statistics Status and Info
- 6. Table Performance I/O and Info
- 7. Index Performance I/O and Usage
- 8. Most Costly SQL

New/Updated Sysmaster Scripts

- 01-chunkio.sql
- 01-chunkio_times.sql
- 01-dbspaceio.sql
- 02-checkpoint_last.sql
- 02-checkpoint_sumary.sql
- 03-logs_not_backup.sql
- 03-logs_usage.sql
- 04-Ratios.sql
- 05-tableinfo_upd.sql
- 06-index_use.sql

- 07-aus_last_run.sql
- 07-upstats_info.sql
- 08-sql_statistics.sql
- 08-syssqexplain.sql
- run_db_loop.sh
- upstats_info.sql
- index_use.sql

Disclaimer

- The New scripts are experimental
- Not completed works
- Still under test and development
- Only tested on 14.10.FC1 and 12.10.FC10
- May break and many need to be updated in the future

Past Presentations

- Informix Performance Tuning Using the Sysmaster Database
 - Webcast October 5, 2017 130 slides and scripts
 - Informix Conference 2017, 2016,.... 1997
 - Webcast March 24, 2015 63 slides and scripts
 - Webcast April 30, 2013 101 slides and scripts
- Webcast Replay on YouTube:
 - https://www.youtube.com/user/AdvancedDataTools/videos
 - http://advancedatatools.com/Informix/Webcasts.html
 - http://www.advancedatatools.com/Articles/Sysmaster/Sysmaster.html

Sysmaster - New 2019

Advanced DataTools

7

What is the Sysmaster Database?

A database that peeks into the shared memory structures of an INFORMIX-Dynamic Server

Sysmaster - New 2019

Informix Control Structures in Memory are the Sysmaster Database



Sysmaster Database contains:

- Server information
- Dbspace & chunk information
- Database & table information
- User session information
- Currently running SQL

Performance of queries on Sysmaster Database

The data is in shared memory but:

- Views used by tables require disk access and may be slow
- Complex views used to hide complex data
- Some tables are large (million locks)
- Unbuffered logging of temp tables

Sysmaster - New 2019

Differences from other databases

- Do not update Sysmaster tables as this may corrupt the server
- Cannot use dbschema on pseudo tables
- Cannot drop pseudo tables or the Sysmaster Database

Isolation level is Dirty Read

- Data is dynamic and can change as you retrieve it (Dirty Read)
- Dynamic nature may return inconsistent results
- However, it uses Unbuffered logging and temp tables are logged

Sysmaster Database may change

- Some undocumented tables and columns may change in future versions
- Scripts in this presentation using undocumented features may not work on all versions of Informix
- New scripts have been run on versions 12.10 and 14.10.FC1
- Sysmaster upgrades in 14.10.FC2

Sysmaste 12.10 vs 14.10.FC1

linfor 3c3	mix@tiger1:/opt tra	ain1 > diff in	formix12.10.FX10/etc/sysmaster.sql :	informix14.10.FC1/etc/sysm	aster.sql
< {*	Licensed Materials	s – Property o	f IBM and/or HCL	*}	
 > {*	Licensed Materials	s - Property o	f IBM and/or HCL	*}	
5C5 < {*	IBM Informix Dyna	mic Server		*}	
> {*	IBM Informix Dyna	mic Server		*}	
< {*	(c) Copyright HCL	Technologies	Ltd. 2017. All Rights Reserved.		*}
 > {* 197a1	(c) Copyright HCL	Technologies	Ltd. 2017, 2018. All Rights Reserv	ed. *}	
> 319c3	idx_acc_tm 20.321	integer,	{ Last access time (for indexes)	}	
<	pn_flags2	integer	{ partition flags2	}	
>	pn_flags2 pn_idx_acc_tm	integer, integer	{ partition flags2 { last index access time	} }	
481c4	83 create unique indu	ex informix sv	erstchech on everstch (sch) hash on	(sch) with 2 buckets.	
> 2125a	create index info: 2128,2129	rmix.sysrstcbs	cb on sysrstcb (scb) hash on (scb) y	with 2 buckets;	
>	Note that sysne	tworkio2 will	only be used for equality lookups		
> 2354c	create index info: 2358,2360	rmix.sysnetwor	kio2 on sysnetworkio(sid) hash on (sid) with 2 buckets;	
<	sdscycle_acked	integer			
>	sdscycle_acked	integer,			
>	logid_applied	integer,	{ unique log id of last log page a	pplied}	
>	logpage_applied	integer	{ page number of last log page app	lied }	
2522c	2528				
<	version	char(12),	{ IDS version }		
>	version	char(24),	{ IDS version }		
TULLOL	mixerigeri:/opt tra	aini > is			

How do you measure the time since the Server Statistics have been reset?

Required for Performance Ratios

Interesting undocumented table – Sysshmvals*

sh mode sh boottime sh pfclrtime sh curtime sh bootstamp sh_stamp sh_mainlooptcb sh_sysflags sh maxchunks sh maxdbspaces sh maxtrans sh maxlocks sh maxlogs sh nbuffs sh_pagesize sh nlrus

sh_maxdirty sh mindirty sh ncleaners sh_longtx

int. turbo mode number int, boot time of day

int, time profilers were last clr

int, current mt time

int. boot time stamp

int, current time stamp

int. address of main thread

int, system operating flags

int, size of chunk table

int, size of dbspace table

sh_maxuserthreads int, max # of user structures

int. max # of trans structures

int. # of locks total

int, size of log table

int. # of buffers total

int, buffer size in bytes

int, # of Iru queues

float, LRU max % dirty pages float, LRU min % dirty pages

int, # of cleaning/flushing procs

int, # the long transaction flag

sh_optstgbsnum sh cpflag sh_rapages sh rathreshold sh lastlogfreed sh rmdlktout sh narchivers sh maxpdgpriority int, max pdgpriority sh_fuzcpflag sh_needcpsyn sh nfuzzy sh nfuzzypre sh oldestlsnug sh_oldestlsnpos sh builddpt sh ndptentries sh dptsize sh curmaxcons

sh ovlmaxcons

int, subsystem Blobspace int, TRUE => doing checkpoint int, # pages to read ahead int, # to start next read ahead int, last log (id) written to tape int. max timeout when distributed int. number of active archives int, fuzzy checkpoint flag int, hard checkpoint int, # buffers marked fuzzy int, # buffers fuzzy in last ckpt int, Isn of oldest update not int, flushed to disk int, builing DPT necessary int. # entries in DPT int. size of DPT int. max #connections in this run int max #connections to server

Sysmaster - New 2019

Advanced DataTools

17

When were the Statistics Cleared?

- -- Module: @(#)server_uptime.sql 2.0 Date: 2013/04/10
- -- Author: Lester Knutsen Email: lester@advancedatatools.com
- -- Advanced DataTools Corporation
- Discription: Displays how long the Informix Server has been up and when the —— last time stats (onstat –z) were cleared.
 - Tested with Informix 11.70 and Informix 12.10

database sysmaster;

select

```
current current_time,
DBINFO ('utc_to_datetime', sh_boottime ) boot_time,
DBINFO ('utc_to_datetime', sh_pfclrtime) stats_reset_time,
current - DBINFO ('utc_to_datetime', sh_pfclrtime) interval_since_stats_reset,
( sh_curtime - sh_pfclrtime) units second secounds_since_stats_reset,
(ROUND (( sh_curtime - sh_pfclrtime)/60) ) minutes_since_stats_reset
from sysshmvals;
```

Sysmaster - New 2019

DBspace and Chunks Performance

- Started with a script to measure dbspace free
- Added Summary of underlying Chunk IO
- Added Ratios
 - Real read ahead
 - Pages per read/write operation
 - Pages read per minute
 - Pages write per minute
 - Percent of Total IO
- Data from syschktab and sysdbstab

What Percent of Dbspace is Free?

- -- Module: @(#)dbspace_free.sql 2.5 Date: 2013/04/10
- -- Author: Lester Knutsen Email: lester@advancedatatools.com Advanced DataTools Corporation

database sysmaster;

select name[1,8] dbspace, -- name truncated to fit on one line sum(chksize) Pages_size, -- sum of all chuncks size pages sum(chksize) - sum(nfree) Pages_used, sum(nfree) Pages_free, -- sum of all chunks free pages round ((sum(nfree)) / (sum(chksize)) * 100, 2) percent_free from sysdbspaces d, syschunks c where d.dbsnum = c.dbsnum group by 1 order by 1;

Informix Best Practices

Dbspaceio.sql – 1 of 3

select -- How long has the server been running current current_time, (select DBINFO ('utc_to_datetime',sh_pfclrtime) from sysshmvals) stats_reset_time, (select (ROUND ((sh_curtime - sh_pfclrtime)/60)) from sysshmvals) minutes_since_stats reset, -- Dbspace Information name dbspace, d.pagesize, count (chknum) num_chunks, sum(chksize) size_sys_pages, sum(nfree) free_pages, -- Estimate Size in KB using base page value (sum(chksize) * (select trunc((sh_pagesize /1024)) from sysshmvals)) size_kb, (sum(nfree) * (select trunc((sh_pagesize /1024)) from sysshmvals)) free_kb, -- Base Performance Info sum(pagesread) pagereads, sum(pageswritten) pagewrites, sum(reads) num_reads, sum(writes) num_writes, sum(readtime) usecs_readtime, sum(writetime) usecs_writetime,

Dbspaceio.sql 2 of 3

```
-- Ratio of pages per read (Read ahead)
```

case

```
when sum( pagesread ) = 0 then 0
when sum( reads )= 0 then 0
else
```

```
(sum(pagesread)/sum(reads))
```

end pages_per_read,

-- Ratio of pages per write (Write ahead)

case

```
when sum ( pageswritten ) = 0 then 0
when sum ( writes) = 0 then 0
else
( sum( pageswritten ) / sum( writes ))
end pages_per_writes,
```

Dbspaceio.sql 3 of 3

```
-- Ratio of page reads per minute
```

case

```
when sum( pagesread ) = 0 then 0
```

else

(sum(pagesread) / ((select (ROUND ((sh_curtime - sh_pfcIrtime)/60)) from sysshmvals))) end pages_read_per_minute,

-- Ratio of page writes per minute

case

```
when sum( pageswritten ) = 0 then 0
```

else

(sum(pageswritten) / ((select (ROUND ((sh_curtime - sh_pfcIrtime)/60)) from sysshmvals))) end pages_write_per_minute,

-- Percent of IO of Total

round(sum(pagesread) / (select sum(pagesread) from sysmaster:syschktab) , 2) read_percent,

round(sum(pageswritten) / (select sum(pageswritten) from sysmaster:syschktab) , 2) write_percent sysmaster:syschktab c, sysmaster:sysdbstab d

where c.dbsnum = d.dbsnum

group by 1,2,3,4,5

order by 4 desc;

from

Dbspaceio.sql

sysmaster@train1

current time stats_reset_time minutes_since_sta+ dbspace pagesize num chunks 1 size_sys_pages free_pages size kb free kb pagereads pagewrites num_reads num writes usecs_readtime usecs_writetime pages_per_read pages_per_writes pages_read_per_mi+ pages_write_per_m+ read_percent write_percent 0.45

2019-09-24 20:33:33.000 2019-09-22 12:46:02 3348 datab3dbs 2048 5000000 2647971 10000000 5295942 1147219245 210703061 2352258 130562745 855737009.0778 614901469148.8 487.709785661267 1.61380691712632 342658.077956989 62934.0086618877 0.94

Demo Scripts

- dbspaceio.sql
- chunkio.sql
- chunk_io_times.sql

Chunk_io_times..sql

sysmaster@train1 ----- Pro

sysmaster@train1 ----- Pres

dbspace device chknum reads readtime avg_read_svc_time writes writetime

rootdbs /informixchunks/train1/rootdbs 1 25590 7077048.724443 276.555245191220 722864 3611851623.459

dbspace device chknum 2 reads readtime avg_read_svc_time writes writetime

loadbs /informixchunks/train1/logdbs 3945.029112679 1315.00970422635 2316003 11282877785.61

dbspace device chknum reads readtime writes writetime avg_write_svc_time 376.043681617670

dbspace device chknum reads readtime writes writetime

rootdbs /informixchunks/train1/rootdbs 1 82645 3867209.703473 2547269 957884412.8306

logdbs /informixchunks/train1/logdbs 2 657.8710621879 avg_read_svc_time 164.467765546978 8526352 2428285486.804

Checkpoint Performance

- What is a summary of my Checkpoint Performance?
 - Checkpoint_summary.sql
- What are the details of the last 10 Checkpoints?
 - Checkpoint_last.sql

Syscheckpoint table

Intvl	Internal sequence number of the checkpoint
Туре	Type of checkpoint, Blocking or Non-Blocking
Caller	Reason for checkpoint, CKPTINT, Physical Log, Logical Log, User
clock_time	Time of checkpoint (System time)
crit_time	Time spent performing checkpoint
flush_time	Time spent flushing pages to disk
cp_time	Time spent from checkpoint pending start to complete
n_dirty_buffs	Number of dirty pages to flush
plogs_per_sec	Average number of pages in physical log per second
llogs_per_sec	Average number of pages in logical log per second
dskflush_per_sec	Average number of disk pages flushed per second
ckpt_logid	Logical Log Id of checkpoint
ckpt_logpos	Logical Log position of checkpoint
Physused	Physical Log pages used
Logused	Logical Log pages used
n_crit_waits	Number of critical waiters threads
tot_crit_wait	Time of critical waiters
<pre>longest_crit_wait</pre>	Longest Time of crtical waiters
block_time	Time checkpoint blocked threads

Informix Best Practices

Checkpoint Performance Summary

select

type,

count(*) num_checkpoints, max (dbinfo("utc to datetime", clock time)) last checkpoint, -- Clock time of the checkpoint max (crit_time) max_sec_crit_time, -- Fractional seconds spent in critical sections sum (crit_time) sum_sec_crit_time, -- Fractional seconds spent in critical sections max (flush_time) max_sec_flush_time, -- Fractional seconds spent flushing dirty pages during the checkpoint sum (flush time) sum sec flush time, -- Fractional seconds spent flushing dirty pages during the checkpoint max (cp_time) max_checkpoint_time, -- Duration of the checkpoint in fractional seconds sum (cp_time) sum_checkpoint_time, -- Duration of the checkpoint in fractional seconds max (n_dirty_buffs) max_dirty_buffs, -- Number of dirty buffers at the beginning of the checkpoint sum (n_dirty_buffs) sum_dirty_buffs, -- Number of dirty buffers at the beginning of the checkpoint max (n_crit_waits) max_crit_waits, -- Number of processes that had to wait for the checkpoint sum (n_crit_waits) sum_crit_waits, -- Number of processes that had to wait for the checkpoint max (tot_crit_wait) max_crit_sec, -- Total time all processes waited for the checkpoint - fractional seconds sum (tot_crit_wait) sum_crit_sec, -- Total time all processes waited for the checkpoint - fractional seconds max (block_time) max_block_time, -- Longest any process had to wait for the checkpoint - fractional seconds sum (block_time) sum_block_time -- Longest any process had to wait for the checkpoint - fractional seconds from syscheckpoint group by 1 order by 1;

Checkpoint_summary.sql

sysmaster@train1 -

type

num_checkpoints last_checkpoint max sec crit time sum sec crit time max sec flush time sum sec flush time max_checkpoint_ti+ sum_checkpoint ti+ max_dirty_buffs sum dirty buffs max crit waits sum crit waits max_crit_sec sum_crit_sec max_block_time sum_block_time

Blocking 8 2019-09-24 21:07:41 1.725911e-05 7.05932528e-05 0.001612641024 0.00474524459 0.004824562211 0.014854903223 52 141 1 3 0.003520003761 0.00634974786 0.00 0.00

type

num checkpoints last_checkpoint max sec crit time sum_sec_crit_time max sec flush time 63.46445248515 sum_sec_flush_time 92.86006797244 max checkpoint ti+ 63.48626005307 sum_checkpoint_ti+ 93.01298011093 max_dirty_buffs sum dirty buffs max crit waits sum crit waits max crit sec sum_crit_sec max_block_time sum_block_time

Non-Blocking 20 2019-09-24 20:32:15 3.04870461e-05 0.000438870645 251943 436956 1 3 36.72768298873 36.74681120340 36.72765214286

46.15479190084

sysmaster@train1 -

Last 10 Checkpoints

select first 10

intvl,

type,

dbinfo("utc_to_datetime", clock_time), -- Clock time of the checkpoint crit_time, -- Fractional seconds spent in critical sections flush_time, -- Fractional seconds spent flushing dirty pages during the checkpoint cp_time, -- Duration of the checkpoint in fractional seconds n_dirty_buffs, -- Number of dirty buffers at the beginning of the checkpoint n_crit_waits, -- Number of processes that had to wait for the checkpoint tot_crit_wait, -- Total time all processes waited for the checkpoint - fractional seconds block_time -- Longest any process had to wait for the checkpoint - fractional seconds from syscheckpoint order by intvl desc;

Checkpoint_last.sql

	sysmaster@train1
(count(*))
:	20
intvl	1073
tvpe	Non-Blocking
(expression)	2019-09-24 20:32:15
crit time	1.85065562e-05
_ flush_time	63.46445248515
cp_time	63.48626005307
n_dirty_buffs	251943
n_crit_waits	1
tot_crit_wait	0.015076670648
olock_time	36.72765214286
intvl	1072
type	Non-Blocking
(expression)	2019-09-24 20:30:41
crit_time	6.19374392e-06
flush_time	29.28418556606
cp_time	29.29495350620
n_dirty_buffs	184573
n_crit_waits	1
tot_crit_wait	0.004051544024
olock_time	9.427139757984

Logical Log Performance

- For HDR Planning How much data will be going to the Secondary Servers?
- What is my Log turnover rate?
- Do I have enough Logs?
- Are the Logs too small or too big?
- Goal Enough Logs for 4 days
- Goal Turnover 12 to 24 Logs per hour

Logical Log Performance

```
"Logs for last 7 days",
select
     count(*) logs used,
     sum(size) log_pages_used,
     dbinfo('utc_to_datetime', min(filltime)) start_time,
     dbinfo('utc to datetime', max(filltime)) end time,
     (dbinfo('utc to datetime', max(filltime)) - dbinfo('utc to datetime', min(filltime))) total time,
     ((max(filltime)) - (min(filltime))) total_secs,
     ((( max( filltime)) - ( min( filltime) )) /60 ) total_minutes,
     (((( max( filltime)) - ( min( filltime) )) /60 ) /60 ) total_hours,
     (count(*) / ((((max(filltime)) - (min(filltime))) /60) /60)) logs_per_hour,
     (sum(size) / ((((max(filltime)) - (min(filltime))) /60) /60)) pages_per_hour
from syslogfil
where filltime > 0
and (dbinfo('utc to datetime', (filltime)) > (current - 7 units day));
```

Logs_usage.sql

a hada da	sysmaster@train1	Press			sysmaster@train1	Press
(constant) logs_used log_pages_used start_time end_time total_time total_secs total_minutes total_hours logs_per_hour pages_per_hour	Logs for last 7 days 79 790000 2019-09-24 21:00:00 2019-09-24 21:07:32 0 00:07:32 452 7.5333333333333 0.125555555555 629.2035398230 6292035.39823009		(c lo st en to to lo pa	onstant) gs_used g_pages_used art_time d_time tal_time tal_secs tal_minutes tal_hours gs_per_hour ges_per_hour	Logs for last 7 days 79 790000 2019-09-24 13:46:19 2019-09-24 20:37:29 0 06:51:10 24670 411.166666666667 6.8527777777778 11.52817186867 115281.718686664	
number fillt	ime (expression)	size	nu	mber fillt	ime (expression)	size
1 1569373 2 1569373 3 1569373 4 1569373 5 1569373 6 1569373 17 1569373 18 1569373 19 1569373 20 1569373	528 2019-09-24 21:05:28 528 2019-09-24 21:05:28 528 2019-09-24 21:05:28 529 2019-09-24 21:05:29 529 2019-09-24 21:05:29 529 2019-09-24 21:05:29 529 2019-09-24 21:05:29 529 2019-09-24 21:05:30 530 2019-09-24 21:05:30 530 2019-09-24 21:05:30 530 2019-09-24 21:05:30 531 2019-09-24 21:05:31	10000 10000 10000 10000 10000 10000 10000 10000 10000		1 1569371 2 1569371 3 1569371 4 1569371 5 1569371 6 1569371 17 1569371 18 1569371 19 1569371	409 2019-09-24 20:30:09 410 2019-09-24 20:30:10 411 2019-09-24 20:30:11 412 2019-09-24 20:30:12 414 2019-09-24 20:30:14 416 2019-09-24 20:30:16 420 2019-09-24 20:30:20 425 2019-09-24 20:30:25 430 2019-09-24 20:30:30	10000 10000 10000 10000 10000 10000 10000 10000

Logical Log Not Backed up

select uniqid, is_current, is_used, is_backed_up, is_new
from syslogs
where is_used = 1
 and is_new = 0
 and is_temp = 0
 and is_pre_dropped = 0
 and is_backed_up != 1
order by uniqid;

Logs_not_backup.sql

	<u></u>	sysmaster@ti	rain1	Press CTRL-W	for H
uniqid	is_current	is_used	is_backed_up	is_new	
7271	1	1	0	0	

Server Ratios Dashboard

- Goal One SQL Script to Show a Status of the Server
 - Ratios.sql
- Based on a Union of 16 SQL Scripts

Sysprofile (onstat -p)

View sysprofile: Current statistics and performance information of the server.

name	char(32),	profile	element	name
value	int8	current	value	

The values are re-set to 0 when Informix is shutdown and started and when the command "onstat -z" is used.

Sysmaster - New 2019

Sysprofile = onstat -p

IBM Informix Dynamic Server Version 12.10.FC9 -- On-Line (CKPT INP) -- Up 09:51:00 -- 14723064 Kbytes Blocked:CKPT

Profile dskreads pagreads bufreads %cached dskwrits pagwrits bufwrits %cached 1018991690 1350945127 7538138566 86.49 121620012 174286092 963734463 87.38 isamtot read write rewrite delete commit rollbk start open 10313914398 6008836 359791 1869291677 595314515 83203954 1833369 4236906 32935 gp_write gp_del gp_alloc gp_read gp_rewrt gp_free gp_curs 0 0 0 0 0 0 0 ovlock ovuserthread ovbuff numckpts flushes usercpu syscpu 72322.30 15949.95 373 1240 0 0 0 lokwaits bufwaits lockregs deadlks dltouts ckpwaits compress segscans 192124 12173696 5803250 6782332472 0 4594872 10810 0 idx-RA ixda-RA da-RA logrec-RA RA-pgsused lchwaits 923776263 625668471 413179008 573414996 432062 5334094

Sysmaster - New 2019

Sysprofile

name	value
dskreads	1018537991
bufreads	7522413742
dskwrites	121271673
bufwrites	961215335
isamtot	10296434334
isopens	5976171
isstarts	358703
isreads	1862764237
iswrites	593583519
isrewrites	82910755
isdeletes	1822514
iscommits	4212939
isrollbacks	32767
ovlock	0
ovuser	0
ovtrans	0
latchwts	625532480
buffwts	12154230
lockreqs	6749776961
lockwts	5769870
ckptwts	190970
deadlks	0
lktouts	0
numckpts	372
plgpagewrites	20335533
plgwrites	318371
llgrecs	289177909
llgpagewrites	34970632
llgwrites	2419517
pagreads	1350419379

Sysmaster - New 2019

Server Ratios Dashboard

- sysmaster@train1 ----

metric

Statics Uptime in Hours: Statics Uptime in Minutes: Read Ahead Ratio: Total Sequencial Scans: Scans per hour: Total Sorts: Memory Sorts: Disk Sorts: Max Sort Space: Sorts per hour: Buffer Reads per hour: Buffer Writes per hour: Commits per hour: Buffer Waits per hour: Checkpoints per hour: Lock Requests: Lock Waits: Dead Locks: Lock Wait Ratio Foreground Writes (Very Bad): 2.00000000000000 LRU Background Writes: Chunk Writes: LRU to Chunk Writes:

106.05000000000 6363.0000000000 0.37652675893228 71051.0000000000 670.292452830189 63210.000000000 52268.000000000 10942.0000000000 150784.000000000 596.320754716981 258683393.594340 134273634.867925 2148.67924528302 1543.24528301887 12.4433962264151 5488682520.00000 2110.0000000000 0.00000000000000 2601271.33649289 562075525.000000 773271077.000000 0.72688031625422

value

sysmaster@train1

metric

Statics Uptime in Hours: Statics Uptime in Minutes: Read Ahead Ratio: Total Sequencial Scans: Scans per hour: Total Sorts: Memory Sorts: Disk Sorts: Max Sort Space: Sorts per hour: Buffer Reads per hour: Buffer Writes per hour: Commits per hour: Buffer Waits per hour: Checkpoints per hour: Lock Requests: Lock Waits: Dead Locks: Lock Wait Ratio Foreground Writes (Very Bad): 10583.0000000000 LRU Background Writes: Chunk Writes: LRU to Chunk Writes:

55.88333333333333 3353.0000000000 0.53683721871410 24163.000000000 431.482142857143 21559.000000000 18594.000000000 2965.0000000000 150144.000000000 384.982142857143 127297929.017857 68602726.1071429 1282.42857142857 3123.64285714286 18.8035714285714 1458102556.00000 3,00000000000000 0.00000000000000 486034185.333333 208151173.000000 172388103.000000 1.20745671759031

value

Update Statistics Status

 When did Automatic Update Status Last Run?

 When did Update Status Last Run for all database and tables

Aus_last_run.sql

	benchmark3@ [.]	train1	Press CTRL-W for Hel
db	table	leve	l when
benchmark1	state	1	2019-09-24 01:01:16
benchmark1	state	Н	2019-09-24 01:01:16
benchmark1	benchmark	1	2019-09-24 01:01:16
benchmark1	benchmark	Н	2019-09-24 01:01:16
benchmark2	state	1	2019-09-24 01:01:21
benchmark2	state	Н	2019-09-24 01:01:21
benchmark2	customer	1	2019-09-24 01:01:21
benchmark2	customer	Н	2019-09-24 01:01:21
benchmark2	product	1	2019-09-24 01:01:21
benchmark2	product	Н	2019-09-24 01:01:21
benchmark2	bills	1	2019-09-24 01:01:21
benchmark2	bills	H	2019-09-24 01:01:21

Upstats_info.sql

tabname	customer
low update	2019-09-21 01:11:02.00000
column	customer number
unddate	09/21/2019
mode	Ч
undtimo	2010-00-21 01.11.02 00000
uputime	
ирицигаттон	0:00:00.021/5
maxseqno	9
tabname	product
low_update	2019-09-21 01:11:02.00000
column	product_number
upddate	09/21/2019
mode	Н
updtime	2019-09-21 01:11:02.00000
updduration	0:00:00.00001
maxseqno	1
tabname	state
low update	2019-09-21 01:11:02.00000
column	state
unddate	09/21/2019
mode	Н
undtime	2010-00-21 01.11.02 00000
upduration	
	0.00.00.00002
maxseqno	2

Table Performance and Information

- Everything you want to know about your tables
- Load results into an Excel Worksheet
- Updated to exclude System tables and Indexes

Questions about Tables

- What tables fit on a page?
- What tables have free space before a new extent?
- How big are the tables?
- What tables have the most lock activity?
- What tables have the most I/O activity?
- What tables have sequential scans?
- What is the buffer read % by table?
- What tables could be partitioned?

Base Tables

- Systabnames Basic Table Information
- Systabinfo undocumented
- Sysptprof Performance Information

All the Information about a Table

```
Module: @(#)tableinfo2016.sql 1.0 Date: 2016/04/01
Author: Lester Knutsen Email: lester@advancedatatools.com
Advanced DataTools Corporation
    Description: New Table Information Script – Unload the output to a file
and the load the results into a worksheet for analysis
Tested with Informix 11.70 and Informix 12.10
database sysmaster;

    unload to tableinfo2016.uld

select
        systabnames.dbsname
                                                  database,
        systabnames.tabname
( dbinfo('dbspace',
                                       e tabname,
, ti_partnum )) dbspace,
        systabnames.partnum,
        ti_rowsize
ti_ncols
ti_nkeys
ti_nextns
                              row_size,
                             num_columns,
num_indexes,
num_extents,
                             page_size,
pages_total,
pages_used,
        ti_pagesize
        ti_nptotal
        ti_npused
       ti_npdata page
(ti_nptotal – ti_n
ti_nrows num_rows,
case
                               pages_data,
ti_npused ) pages_free,
               when ( (ti_pagesize +4) -24) -
else "Row smaller the pagesize"
                                                                                               then "Row larger then pagesize"
                                                                    < ti_rowsize
        end rowfit,
        case
                when ti_rowsize > 0 then
                       trunc
                                 ((ti_pagesize -24) / ti_rowsize )
               else Ø
        end rows_per_page,
        case
               when ti_rowsize > 0 then
   ( ( trunc ((ti_pagesize -24) / ti_rowsize ) ) * (ti_nptotal - ti_npused ) )
                else Ø
        end free_rows,
DBINF0 ('utc_to_datetime', ti_created ) create_date,
        lockwts,
        deadlks,
        lktouts,
        isreads,
        iswrites
        isrewrites,
        isdeletes,
        bufreads,
        bufwrites,
        seascans,
        pagreads,
        pagwrites
          bufreads + bufwrites ) total io,
        case
               when pagreads > 0 then
                          pagreads / bufreads )
                       else
                                0
        end buff_read_percent,
        case
               when pagwrites > 0 then
   ( pagwrites / bufwrites )
   else 0
else 0
end buff_write_percent,
(( ti_npdata * seqscans ) * ti_pagesize ) total_bytes_scanned
from systabnames, systabinfo, outer sysptprof
where systabinfo.ti_partnum = systabnames.partnum
and systabinfo.ti_partnum = sysptprof.partnum
and systabnames.dbsname not in ( "sysmaster", "sysuser", "sysutils", "sysadmin" )
and ti_npdata > 0 -- remove partitions with no data pages
and ti_npdata > 0 -- ro
order by total_io desc;
     Sysmaster - New
                                                                                                                                                   49
                                                   Advanced DataTools
     2019
```

Index Performance and Usage

- When was my index last used?
 - Oncheck –pt database:table
- What is the ration of index reads to writes?
- The more reads... the better an index is used..

New Oncheck to Show Index Last Used – oncheck -pT

Level	Total	Average No. Keys	Average Free Bytes	Average Del Keys
1	1	94	 896	

Index Usage Report for index 104_8 on benchmark2:informix.bills

2 3	94 7861	83 153	1016 18	0	
Total	7956	153	29	0	

Index idx_bills_customer_number fragment partition datadbs in DBspace datadbs

Physical Address	3:571	14.14.25			
	001				
IBLSpace Flags	801	Page Locking			
		IBLspace use	4 bit bit-maps		
Maximum row size	1266				
Number of special columns	0				
Number of keys	1				
Number of extents	24				
Current serial value	1				
Current SERIAL8 value	1				
Current BIGSERIAL value	1				
Current REFID value	1				
Pagesize (k)	2				
First extent size	4				
Next extent size	512				
Number of pages allocated	4096				
Number of pages used	3678				
Number of data pages	0		4		
Number of rows	0				
Partition partnum	3145800				
Partition lockid	3145798				
Last Lookup/Scan	Fri Aug 16	14:37:09 2019		Index L	.ast Used

We will Use Data from 3 Sources

- Sysmaster database
- Sysadmin database
- System Tables in each database

 Important to join between Sysmaster and Systables database using table partno to avoid duplicate data

How to Monitor Index Usage

Compare reads and writes on an index...fewer reads indicates the index may not be needed unless it is constraint My old script – does

not handle Indexes

with the same names

in different databases

Select

a.tabname, b.idxname, bufreads, bufwrites, case

when bufwrites = 0 then bu when bufreads = 0 then 0 else (bufreads /bufwrites)

end ratio

from systables a, sysindexes b, outer sysmaster:sysptprof p where a.tabid = b.tabid and p.tabname = b.idxname and a.tabid > 99;

New - How to Monitor Index Usage

Compare reads and writes on an index by partition ... fewer reads indicates the index may not be needed unless it is a constraint

select

	t.tabname,
	i.indexname,
	bufreads,
	bufwrites,
	case
	when bufwrites = 0 then bufreads
	when bufreads $= 0$ then 0
	else (bufreads /bufwrites)
	end ratio
from	systables t, sysfragments i, outer sysmaster:sysptprof p
where	t.tabid = i.tabid
and	i.fragtype = "I"
and	i.partn = p.partnum
and	t.tabid > 99;
	Advanced NataTools

New - How to Monitor Index Usage

- -- ## Module: @(#)index_usage2.sql 2.0 Date: 08/25/2019
- -- ## Author: Lester Knutsen Email: lester@advancedatatools.com
- -- ## Advanced DataTools Corporation

select

```
t.tabname,
        i.indexname,
        bufreads,
        bufwrites,
        case
                when bufwrites = 0 then bufreads
                when bufreads = 0 then 0
                else ( bufreads /bufwrites )
        end ratio
        systables t, sysfragments i, outer sysmaster:sysptprof p
from
        t.tabid = i.tabid
where
        i.fragtvpe = "I"
and
        i.partn = p.partnum
and
        t.tabid > 99;
and
```

New - How to Monitor Index Usage

tabname
indexname
bufreads
bufwrites
ratio

state idx_state_1 15 11 1.363636363636363636

tabname indexname bufreads bufwrites ratio

zip idx_zip_1 630617 931 677.354457572503

tabname indexname bufreads bufwrites ratio

benchmark idx_benchmark_1 214154 4614 46.4139575205895



The Most Costly SQL Running

- Collect a snapshot of current running SQL and save it (once an hour)
- Summarize and review the results
- Show the Top 10 most Expensive SQL on your system

What is the most expensive SQL running?

- Use SQL Trace Real time capture of the cost of what is running (This is a separate presentation)
- Use the view Syssqexplain to capture what is running now
- Script: syssqexplain.sql

What is the most expensive SQL running?

- Documented View Syssqexplain
- Based on internal table Syssdblock and Sysconblock

View: Syssqexplain



Sysmaster - New 2019

Internal Table: Sysconblock

{ Conblock }							
create table informix	.sysconblock	{ Internal Use Only					
(
cbl_sessionid i	integer,	{ session id	}				
cbl_sdbno i	integer,	<pre>{ position in sdblock array</pre>	}				
cbl_conbno s	smallint,	<pre>{ position in conblock list</pre>	}				
cbl_ismainblock c	char(1),	{ main block for statement?	}				
cbl_selflag s	smallint,	{ see cb_selflag (SQ_*)	}				
cbl_estcost i	integer,	{ see <pre>cb_estcost</pre>	}				
cbl_estrows i	integer,	{ see cb_estsize	}				
cbl_flags i	integer,	{ see cb_flags	}				
cbl_flags2 i	integer,	{ see cb_flags2	}				
cbl_seqscan s	smallint,	{ # of SEQUENTIAL SCANs	}				
cbl_srtscan s	smallint,	{ # of SORT SCANs	}				
cbl_autoindex s	smallint,	{ # of AUTOINDEX PATHs	}				
cbl_index s	smallint,	{ # of INDEX PATHs	}				
cbl_remsql s	smallint,	{ # of REMOTE PATHs	}				
cbl_mrgjoin s	smallint,	{ # of MERGE JOINs	}				
cbl_dynhashjoin s	smallint,	{ # of DYNAMIC HASH JOINs	}				
cbl_keyonly s	smallint,	{ # of (Key-Only)s	}				
cbl_tempfile s	smallint,	{ # of Temporary Files	}				
cbl_tempview s	smallint,	{ # of Temp Tables For View	}				
cbl_secthreads s	smallint,	{ # of Secondary Threads	}				
cbl_stmt c	char(32000)	{ current statement	}				
)•							

Sysmaster - New 2019

What is the most expensive SQL running?

- -- Module: @(#)syssqexplain.sql 1.0 Date: 2015/03/20
- —— Author: Lester Knutsen Email: lester@advancedatatools.com
 —— Advanced DataTools Corporation
- -- Description:
 - Tested with Informix 11.70 and Informix 12.10

```
database sysmaster;
```

```
select
```

sqx_estcost, sqx_sqlstatement from syssqexplain into temp A;

```
select
```

sqx_sqlstatement sqlstatement, sum(sqx_estcost) sum_estcost, count(*) count_executions

from A group by 1 order by 2 desc;

Sysmaster - New 2019

Syssqexplain.sql

	sysmaster@train1 Pr
username	informix
sqx_sessionid	15865
sqx_conbno	1
sqx_iscurrent	Y
sqx_estcost	107552
sqx_estrows	675
sqx_sqlstatement	insert into bills (
	customer_number,
	last_name,
	first_name,
	address,
	city,
	state,
	state_name,
	zip,
	start_date ,
	bill_date ,
	product_code ,
	product_number,
	product_name ,
	product_price ,
	product_discount ,
	sales_tx,
	total_bill
	Select
	customer.customer_number,

Questions?



Send follow-up questions to Lester@advancedatatools.com

Sysmaster - New 2019

Advanced DataTools Sessions at IIUG World 2019

Mon 9/23	10:15 AM	C01. Getting Going With Informix Connection Manager	Thomas Beebe
	11:30 AM	A02. Stories from Database Support Customers	Mike Walker
	11:30 AM	D02. Configuring and using the Informix Warehouse Accelerator	Art Kagel
	1:30 PM	B03. Do Stats Better	Art Kagel
Tue 9/24	11:30 AM	B07. Do Storage Better	Art Kagel
	2:45 PM	C09. Setting up SSL for Informix	Thomas Beebe
Wed 9/25	9:30 AM	C11. Migrating your Informix Instance	Mike Walker
	10:15 AM	A12. Exploring the Sysmaster - my new stuff	Lester Knutsen
	2:45 PM	B15. Set up a Raspberry Pi	Mike Walker

Free Informix Webcasts

from the IBM Informix Champions

- SQL Explain Using the SQL Optimizer Query Explain Plan, by Lester Knutsen
 - Thursday, October 31, 2019 at 2:00pm EDT
- Update Statistics Best Practices for Informix DBAs, by Lester Knutsen
 - Thursday, November 21, 2019 at 2:00pm EDT
- Coming in 2020 Informix Tutorials Webcast Series
 - One FREE training tutorial a month

Registration and more information: https://advancedatatools.com/Informix/NextWebcast.html

Sysmaster - New 2019

Informix Training Updated for Informix 14.10

Attend classes online on the web or in person at our training center in Virginia. All you need is a web browser to connect to our WebEx training system, and an SSH client (like Putty) to connect to our training lab for hands-on exercises. Each student uses an 8-core Linux server, with 16GB RAM, SSD drives with Informix 14, and several large databases for benchmark exercises.

> October 7-10, 2019 - Informix for Database Administrators

This course is for new database administrators, programmers, and technical support personnel who will be setting up, managing, and tuning IBM Informix databases.

Informix Training in 2020

- May 18-21, 2020 Informix for Database Administrators
- July 6-9, 2020 Advanced Informix Performance Tuning
- October 5-8, 2020 Informix for Database Administrators More information and registration at:

http://www.advancedatatools.com/Training/InformixTraining.html

Informix 14 Training



Each student in class will have a server running Informix 14.10 with:

- 8 CPU Cores
- 16 GB RAM
- 1 SSD Disk
- 1-4 Disks

Class size is limited to 8 students

Attend online or in person!



Informix Support and Training from the Informix Champions!

Advanced DataTools is an Advanced Level IBM Informix Data Management Partner, and has been an authorized Informix partner since 1993. We have a long-term relationship with IBM, we have priority access to high-level support staff, technical information, and Beta programs. Our team has been working with Informix since its inception, and includes 8 Senior Informix Database Consultants, 4 IBM Champions, 2 IIUG Director's Award winners, and an IBM Gold Consultant. We have Informix specialists Lester Knutsen and Art Kagel available to support your Informix performance tuning and monitoring requirements!

- Informix Remote DBA Support Monitoring
- Informix Performance Tuning
- Informix Training
- Informix Consulting
- Informix Development

Free Informix Performance Tuning Webcast replays at:

http://advancedatatools.com/Informix/Webcasts.html Email: info@advancedatatools.com Web: http://www.advancedatatools.com



Thank You



Lester Knutsen Advanced DataTools Corporation

lester@advancedatatools.com

For more information: http://www.advancedatatools.com



