Informix Performance Tuning Take II

By Art Kagel

The Webcast will start soon!

Informix Performance Tuning Take II

By Art Kagel

Art Kagel, Principal Consultant of Advanced DataTools Corporation. Art is a member of the IIUG Board of Directors and a recipient of the IIUG Directors Award. Art is a five time recipient of the IBM Information Champion Award

The Webcast will start soon!

Agenda

- Table Level statistics and data distributions
- Fragment Level statistics and data distributions
- UPDATE STATISTICS FOR PROCEDURE/FUNCTION
- Providing data distributions for User Defined Types
- Dostats and the art of balancing distribution quality with runtime performance
- Other useful utilities in the utils2_ak package

Table Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS LOW gathers data for:
  - Systables
    - Nrows - # rows in the table
    - Npused - # pages occupied by data
    - Ustlowts – time stats were run

Table Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS LOW gathers data for:
  - Syscolumns
    - Colmin – 2nd lowest value – only 1st 4 bytes
    - Colmax – 2nd highest value

Table Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS LOW gathers data for:
  - Sysindices
    - Clust - % of data still clustered
    - Levels – depth of the index
    - Leaves – width of the bottom leaf level
    - Nunique - # unique key values
    - Nrows - # rows recorded
    - Ustlowts – time stats were run

Table Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS MEDIUM gathers data for:
  - IO counts since table creation as of the command
  - Data distributions or histograms based on a sampling algorithm
- You can control the quality of the MEDIUM distributions:
  - RESOLUTION – Default 2.5% - 40 buckets
  - CONFIDENCE – Default 95%
  - SAMPLING – Default – calculated from confidence

Table Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS HIGH gathers data for:
  - IO counts since table creation as of the command
  - Data distributions or histograms by processing every index node or data row in the table
- You have some control over the quality of the HIGH distributions:
  - RESOLUTION – Default 0.5% - 200 buckets

Fragment Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS LOW gathers data for:
  - Sysfragments
    - For table fragments:
      - Npused - # pages of data in the fragment
      - Nrows - # rows in the fragment
    - For index fragments:
      - Clust - % rows still clustered in the fragment
      - Levels – depth of the index fragment

Fragment Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

- UPDATE STATISTICS MEDIUM and HIGH optionally gather data at the fragment level if the table's STATLEVEL attribute is set to FRAGMENT or AUTO (and the criteria are met).
  - For data distributions
  - IO counts since table creation as of the command

Fragment Level Statistics and Distributions

What does the engine do when you UPDATE STATISTICS?

Fragment level distributions are calculated if:

1. The table's partitioning scheme is expression, interval, list, or rolling window based
2. The SYSSBSPACENAME onconfig parameter is set to a valid sbspace
3. If STATLEVEL is set to AUTO and the table has more than 1 million rows

Auto Statistics Mode

- Not AUS!
- New in 11.70
- ONCONFIG parameter AUTO_STAT_MODE controls whether statistics are always gathered or only when criteria are met!
- Default is 1 - enabled
- Set to '0' to disable. - Same as adding the FORCE option to UPDATE STATISTICS. Always calculate distributions.

Auto Statistics Mode

- You can override the AUTO_STAT_MODE setting when running UPDATE STATISTICS by adding an option:
  - FORCE – update distributions regardless of the default auto setting
  - AUTO – update obeying the STATCHANGE pct regardless of the default setting

Auto Statistics Mode

STATCHANGE parameter in the ONCONFIG file controls the selection criteria used by default when AUTO_STAT_MODE is enabled

STATCHANGE pct is compared to the number of data pages inserted, deleted, or updated since the last time statistics were gathered as a percentage of the number of data pages in the table or fragment.

Auto Statistics Mode

- Each table can have its own STATCHANGE attribute value which can be set using:

ALTER TABLE <tab>
    STATCHANGE <%chg | AUTO>;

- Default is “AUTO” which uses the value set in the ONCONFIG file.
- %chg is 0 – 100%. 0=Always.

Best Statistics Ever!

The best statistics run possible is HIGH on every column in the table individually with fragment level distributions enabled for partitioned tables and with as many buckets (100/RESOLUTION) as there are unique values for each column.

SLOWest Statistics Ever!

The slowest statistics run possible is HIGH on every column in the table individually with fragment level distributions enabled for partitioned tables and with as many buckets (100/RESOLUTION) as there are unique values for each column.

Zen and the Art of Statistics Gathering

Like all things in life,

when generating data distributions

we strive for balance!

Ideal balance

Ideal balance in Informix database statistics generation is achieved by following the recommendations in the Informix Performance Guide manual.

Ideal balance

- HIGH for the leading column of each index
- HIGH for the first column from each index that are different between indexes that start with the same columns

Ideal Balance

- Optionally:
  - HIGH on columns used in a partitioning scheme
  - HIGH on columns used as major filters and join conditions
- Dostats cannot currently implement these options – future feature.
- AUS does not implement these options.

Ideal balance

- MEDIUM for all columns that do not qualify for HIGH

Ideal Balance

- UPDATE STATISTICS LOW on each full index key for all indexes on the table.

Ideal Balance

- UPDATE STATISTICS FOR ROUTINE

Recompile all stored procedures and functions after updating statistics for objects they may reference.

*Always SET PDQPRIORITY 0; before recompiling stored procedures and functions! Routines will always execute with the PDQPRIORITY they were compiled with if it is non-zero!

AUTO UPDATE STATISTICS (AUS)

Informix v11.xx and later includes, as one of the tasks delivered with the task manager, two tasks that implement nightly automatic update statistics processing so you don't have to do anything to maintain your statistics and data distributions.

AUTO UPDATE STATISTICS (AUS)

Problems with AUS:

1. The AUS Evaluator task is slow and uses high levels of CPU and IO resources.
(Improved in v12.10 somewhat)
2. The default scheduling could result in the evaluator not completing before the AUS Refresh task starts

AUTO UPDATE STATISTICS (AUS)

Problems with AUS:

3. Does not do LOW on each full index key
4. No distributions on non-indexed columns. This means that if you filter on a non-indexed column the optimizer has no idea how good or bad that filter is and may join tables in the wrong order.

AUTO UPDATE STATISTICS (AUS)

Problems with AUS:

5. Does not recompile stored procedures. (Left to auto-compile the next time they are run.)
6. Not simple to configure except by using OAT.

AUTO UPDATE STATISTICS (AUS)

Problems with AUS:

7. Few control options beyond runtime scheduling:
   - AUS_AGE – define stale stats age
   - AUS_CHANGE – define change % - doesn't override STATCHANGE so setting it lower is no-op
   - AUS_SMALL_TABLES - #rows that defines a small table
   - AUS_AUTO_RULES -
     - 0 – rerun existing stats
     - 1 – calculate new stats using rules

Myschema to the rescue

My myschema utility has an option, --distributions-file=<filepath> that will generate a set of update statistics commands to duplicate the existing stats and distributions. Same as setting AUS_AUTO_RULES to 0 and running AUS.

Dostats to the rescue

My dostats utility faithfully implements the recommendations in the performance guide optimized for best performance.

Dostats to the rescue

Options to:

- Force override STATMODE
- Obey STATLEVEL & other AUS settings
- Select only tables with stale stats (> n days old)
- Select only tables with change in nrows > x%
- Process or ignore lists of tables
- Process all databases
- Include catalog tables
- Report run times for each operation and summary level
- Report clock time for each operation and summary level
- Adjust RESOLUTION, CONFIDENCE, & SAMPLING

Dostats to the rescue

Options to:

- Process commands immediately
- Write commands to an SQL file to process later
- Install a stored procedure to process commands
- Schedule the procedure to run once later or periodically (ie replace AUS)
- Drop all distributions before starting (needed after a server upgrade)
- Drop distributions for each table before beginning processing for each.
- Specify maximum error count to ignore
- Dostats version for older engines (prior to 10.00) included.

Dostats to the rescue

You can replace AUS (both the Evaluator and the Refresh tasks) with dostats. See the May 5, 2011 issue of my BLOG for detailed instructions:

http://informix-myview.blogspot.com/2011/05/aus-versus-dostats.html

Examples of using dostats:

Other utilities in utils2_ak

myschema

Replacement for all dbschema functions except -hd (data distribution report) with additional features:

- Prettier formatting
- Explicitly create implicit indexes before defining constraints so they have usable names.
- Adjust first and next extent sizes to:
  - Current extents
  - Current pages
  - Percent of existing/current
  - Minimum required
  - Min/Max/Avg fragment size
- Suppress/Set object owners
- Order objects based on dependencies, name, or create order

Other utilities in utils2_ak

myschema

- Separate table DDL from other statements in multiple files
- Separate privilege statements to a separate file
- Add conditional creation syntax
- Generate dbimport compatible schema file
- Generate file of external directives only
- Generate file of API commands to defragment tables
- Add ONLINE keyword to CREATE INDEX statements
- Generate infrastructure scripts for versions prior to 11.70
- Generate a script to duplicate existing UPDATE STATISTICS levels.

Other utilities in utils2_ak

dbdelete

Fastest way to delete large numbers of rows from a table.

Avoids long transaction rollbacks with partial commits.

Low overhead.

Works best with non-fragmented tables or fragmented tables created WITH ROWIDS

Locks and commits a limited number of rows at a time to reduce concurrency problems. Adjustable from 512 → 8192 rows by adjusting the transaction buffer size.

Efficient enough to run divide and conquer.

Other utilities in utils2_ak

dbcopy

Extremely fast, low overhead data copy utility. Features:

- Can run at least one copy per CPU VP to divide and conquer.
- Partial commits to avoid long transaction rollbacks.
- Can copy between logged and unlogged databases
- Copy WITHOUT REPLICATION
- Logs insert errors in UNLOAD format for manual correction and reload.
- Copy within a single database/server or between databases/servers.

Other utilities in utils2_ak

dbcopy

Notes:

- Doesn't handle UDTs, CLOB, or BLOB
- Chokes if it has to copy more than one TEXT and/or BYTE column
- Bug in IDS v11.50.xC5+ and CSDK v2.50.xC5+ causes bogus -1831 error if the table has any LVARCHAR columns. Affects any process using fetch array code on LVARCHAR data.
  - Not yet fixed as of v12.10xC2 & CSDK 4.10xC2

Other utilities in utils2_ak

dbmove

New data copy utility. Uses the algorithms from dbdelete to copy and commit blocks of rows using INSERT INTO ... SELECT ... FROM … with a ROWID list

- Works with BLOB, CLOB, and UDT columns
- Works with multiple BYTE and TEXT columns in a record
- Avoids the fetch array with LVARCHAR -1831 bug

Other utilities in utils2_ak

drive_dostats

Distribute tables across <N> copies of dostats running in parallel.

Options to:

- Process tables in largest first or smallest first order
- Exclude tables listed on the command line or in a file
- Include only tables listed on the command line or in a file
- Most dostats options passed through

Other utilities in utils2_ak

ul / bload / bunload

Bunload - “ul -u” - Unload data to a file in binary format

Bload – “ul -l” - Load a binary data file to a table

Options:

- Use Buffered, Unbuffered, Asynchronous I/O
- Insert cursor with partial commits
- Insert WITHOUT REPLICATION
- Specify commit size
- Swab data to portable format / NoSwab
    - NoSwab mode is compatible with INFORMIX mode external tables for both loading and unloading.
    - Swab format files are portable between processor types.

Other utilities in utils2_ak

dbstruct

```
$ dbstruct -d art -t syscolumns

typedef struct SYSCOLUMNS_S {
char colname[129];
int tabid;
short colno;
short coltype;
short collength;
int colmin;
int colmax;
int extended_id;
int seclabelid;
short colattr;
} syscolumns_t;
```

Other utilities in utils2_ak

dbstruct

```
$ dbstruct -d art -t syscolumns -e
EXEC SQL BEGIN DECLARE SECTION;

typedef struct SYSCOLUMNS_S {
string colname[129];
int tabid;
short colno;
short coltype;
short collength;
int colmin;
int colmax;
int extended_id;
int seclabelid;
short colattr;
} syscolumns_t;
EXEC SQL END DECLARE SECTION;
```

Other utilities in utils2_ak

dbstruct

```
$ dbstruct -d art -t syscolumns -G

DEFINE syscolumns_rec RECORD
    colname VARCHAR(128),
    tabid INT,
    colno SMALLINT,
    coltype SMALLINT,
    collength SMALLINT,
    colmin INT,
    colmax INT,
    extended_id INT,
    seclabelid INT,
    colattr SMALLINT
END RECORD
```

Other utilities in utils2_ak

dbstruct

```
$ dbstruct -d art -t syscolumns --F

      structure/SYSCOLUMNS_t/
         character*128 colname
         integer*4 tabid
         integer*2 colno
         integer*2 coltype
         integer*2 collength
         integer*4 colmin
         integer*4 colmax
         integer*4 extended_id
         integer*4 seclabelid
         integer*2 colattr
      end structure
      record/syscolumns_t/ syscolumns
      common/syscolumns_c/ syscolumns
```

Other utilities in utils2_ak

sqlstruct

```
$ sqlstruct -n systables -d art
> select tabname, owner, tabid, tabtype from systables;

typedef struct systables_s {
    char tabname[129];
    char owner[33];
    int tabid;
    char tabtype[2];
} systables_t, *systables_tp;
```

Other utilities in utils2_ak

dbsavail

$ dbsavail

Dbspace	Total Pages	Free Pages	Total KB	Free KB	
dumbblobspace	1024000	499	2097152000	1021952	Blob(BlbPg: 4096K)
s9_sbspc	30000	1371	60000	2742	SBsp(PgSz: 2K)
sbspacetemp	50000	2522	100000	5044	TSBs(PgSz: 2K)
sbspace	505000	4292	1010000	8584	SBsp(PgSz: 2K)
plogdbs	81742	10839	163484	21678	(PgSz: 2K)
adtc_upgrd_dbs	50000	22447	100000	44894	(PgSz: 2K)
llogdbs	300753	30700	601506	61400	(PgSz: 2K)
datadbs2	50000	33208	100000	66416	(PgSz: 2K)
tempdbs1	56637	56426	113274	112852	Temp(PgSz: 2K)
datadbs	767498	83998	1534996	167996	(PgSz: 2K)
space16k	800000	377952	12800000	6047232	(PgSz: 16K)
rootdbs	1278780	413293	2557560	826586	(PgSz: 2K)
Totals:	4994410	1037547	2116292820	8387376	

Other utilities in utils2_ak

dbscript

```
$ dbscript -d art -t '*' -c 'alter table %s statchange 10;'
alter table foo statchange 10;
alter table regresstab statchange 10;
alter table bitarraytab statchange 10;
alter table tst statchange 10;
alter table table1 statchange 10;
alter table pagecounts statchange 10;
alter table tab1 statchange 10;
alter table tab2 statchange 10;
alter table char_to_blob_blob statchange 10;
alter table extent_test statchange 10;
alter table systables_private statchange 10;
alter table regresstab_copy statchange 10;
alter table tst_unload statchange 10;

    ...
```

Other utilities in utils2_ak

listdb7

```
$ listdb7 -D
There are currently 20 databases available:

  # Database/Table/Index/Owner        DBSpace/Created  Log/Lock Mode  GLS?
=== ================================ ============== ============= ====
  1 adtc_monitoring
      informix                          rootdbs
                                     07/26/2011     UNBUFFERED     No
  2 art
      informix                          rootdbs
                                     07/26/2011     UNBUFFERED     No
  3 benchmark
      art                            datadbs
                                     04/10/2012     UNBUFFERED     No
  4 bids
      art                            datadbs2
                                     02/13/2012     UNBUFFERED     No
  5 bidw_last
      art                            datadbs2
                                     02/09/2012     UNBUFFERED     No
```

Listdb5 included for OnLine v5.xx engines.

Other utilities in utils2_ak

listdb7

```
listdb7 -D -d art -t
There is currently 1 matching database:

  # Database/Table/Index/Owner         DBSpace/Created  Log/Lock Mode  GLS?
=== ================================= ============== ============= ====
  1 art
      informix                          rootdbs
                                      07/26/2011      UNBUFFERED     No
        a
          art                           rootdbs
                                        03/09/2012  Row
          Idx:  301_1983
                                         rootdbs
              1 Extents 8 Pages total, Average extent: 8 pages.
        a_raw_table
          art
                                        07/26/2011  Page
              1 Extents 8 Pages total, Average extent: 8 pages.
        again
          art
                                        07/26/2011  Page
              1 Extents 8 Pages total, Average extent: 8 pages.
```

Other utilities in utils2_ak

printfreeB

```
$ printfreeB art systables
Looking at DB: art, Table: systables.

Report for table: art:systables in dbspace #1: rootdbs.

    Table partition header reports that table has:
500% free
64 pages allocated in 2048 extents
37 pages used
262 rows of data in 21 data pages
    Sysptnext reports:          64 pages in 4 extents.
    Bitmap scan reports:
Unused pages:                   26.
Bitmap pages:                    1.
Unused blob pages:          0.
Partial data pages:         21.
Partial blob pages:         0.
Small data pages:           0.
Half full blob pages:            0.
Full data pages:            0.
Full index pages:           16.
Full blob pages:            0.
-----------
Total pages reported:           64.
```

Questions?

Next Webcast

- **Date:** February 20, 2014
- **Time:** 2:00pm EST

- ***Accelerating Data Warehouse Queries with Yellowfin BI and the Informix Warehouse Accelerator***

Advanced Informix Training

Advanced Informix Performance Tuning.

Do you want to learn how to be the ***FASTEST Informix DBA***?

This 4-day course focuses on techniques for optimizing an Informix Database. Labs will demonstrate more than 100% performance improvement. Each student will have a 4-core Linux server with IDS 11.70 or 12.10 and a large dataset for benchmark exercises and labs.

Attend the class in person or online via our Web Training Center.

Both Art Kagel and Lester Knutsen will be teaching the Advanced Classes together. (Cost $2,000 USD)

- 2014 Schedule is always on our web site.

Call 703-256-0267 x101 for more information and to Register or email info@advancedatatools.com

Informix Training in 2014

- **Informix for Database Administrators**
  - January 27-30, 2014
  - September 15-18, 2014
- **Advanced Informix Performance Tuning**
  - March 10-13, 2014
  - July 14-17, 2014
- All courses can be taken online on the web from your home or desk or at our training center in Virginia.
- We guarantee to *NEVER* cancel a course and will teach a course as long as one student is registered!

Advanced DataTools

Building a successful data warehouse requires the commitment of significant time and financial resources. To ensure that this commitment is justified, Advanced DataTools has developed a unique approach based on years of designing and implementing successful programs and applications.

Thank You

Art Kagel
Advanced DataTools Corporation

art@advancedatatools.com

For more information:

http://www.advancedatatools.com