ISUG-TECH 2015 Conference

SAP IQ Best Practices for DBAs and Developers

Mark Mumy, SAP

Agenda

- Welcome
- Speaker Introduction
- SAP Big Data Strategy with SAP IQ
- Q&A

Who is Mark Mumy?

- Came to SAP via the Sybase acquisition
- 19+ years with SAP
- Over 19 years experience with SAP IQ, 4+ years with HANA, 3+ years with Big Data
- Dabbled in replication, streaming, business intelligence, ETL
- Been involved in at least half of the SAP IQ architectures
- Author of numerous whitepapers and many TechWave, TechEd, d-code, ISUG Tech sessions
- Spent most of my time focusing on EDW, data marts, ODS, Big Data, high speed/high throughput database computing
- Chief architect on the Sybase/SAP IQ Guinness World Record systems
- Don't hesitate to contact me! Email me at ***mark.mumy@sap.com***

Agenda

Setup and Administration

- Hardware Configuration
- Creating the Database
- Sybase IQ Server Settings
- Connectivity
- Maintenance Tasks
- Memory Usage

Development and Design

- Data Model Recommendations
- Database Programming
- Data Manipulation

SETUP AND ADMINISTRATION

Hardware Configuration

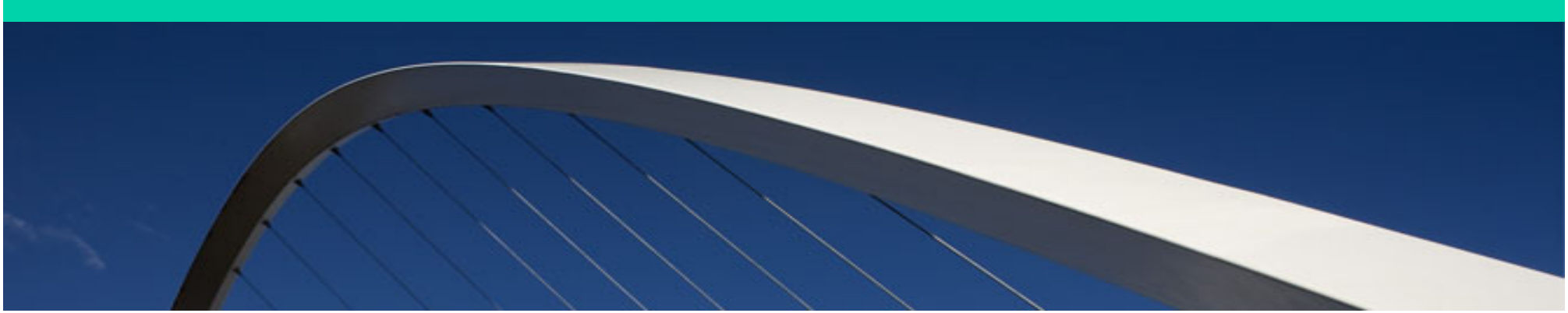
Creating the Database

Sybase IQ Server Settings

Connectivity

Maintenance Tasks

Memory Usage



Hardware Configuration

HARDWARE CONFIGURATION

DBSPACE and DBFILE terminology

- A *dbfile* is what we used to think of as a dbspace
  - Maps to a raw device or filesystem device
  - A physical storage unit
- A *dbspace* is a logical grouping of *dbfiles*
  - Objects (tables, columns, indexes) can now be placed on specific dbspaces to isolate or segregate workload
  - A logical storage unit
- IQ 15 uses the following dbspaces "out of the box"
  - SYSTEM – SQL Anywhere catalog file (.db file)
  - IQ_SYSTEM_TEMP – temporary storage
  - IQ_SYSTEM_MAIN – system storage area
- With the default licensing, users can allocate one more dbspace to store user data
  - Generally called a user defined dbspace

HARDWARE CONFIGURATION

IQ_SYSTEM_MAIN dbspace

- Should not be used to store any user data
- Think of this as a system area
- Used for transaction shipping between nodes and some node to node communication
- Should be on a dedicated device(s)
- For databases less than 100 GB, it should be 5-10% the size of all user defined main stores with a minimum of 4 GB
- For databases more than 100 GB, it should be 1-2% the size of all user defined main stores with a minimum of 8 GB (16 GB for multiplex)
- When looking at IQ_SYSTEM_MAIN dbspace usage, 20% is reserved automatically for IQ so don't be alarmed

HARDWARE CONFIGURATION

IQ_SHARED_TEMP dbspace

- Cannot be used to store any user data
- Used for data shipping between nodes running in a PlexQ mode (MPP)
- Should be on a dedicated device(s)

HARDWARE CONFIGURATION

- Storage
  - Volume Manager
    - Not a necessary or recommended component for Sybase IQ
    - Can use OS level multipathing to provide controller and SAN path failover and dynamic pathing to storage
    - Additional overhead and software that has no value add to an Sybase IQ installation (can add significant value to operations staff!)
    - Acceptable for a Sybase IQ single node operation where hardware does not have the ability to apply RAID to the disk devices
    - If using volume managers in a multiplex, you must upgrade to the cluster/multi-system aware versions

HARDWARE CONFIGURATION

- Drive Arrays
  - RAID Level
    - Recommend using RAID 5 as a blend between performance, protection, and cost
    - RAID 10 (striping and mirroring) for all out performance
  - Recommend using raw devices (a must for multi-node Sybase IQ)
  - For details on drive array specifications and configurations see the IQ Hardware Sizing Guide

Quick Sizing Reference

Memory

- ✓ 8-12 GB per core for simplex
- ✓ 12-16 GB per core for multiplex
- ✓ Main cache: 30% of total RAM
- ✓ Temp cache: 30% of total RAM
- ✓ Large Memory: 30% of total RAM
- ✓ RLV: allocate as necessary

Disk

- ✓ 50-100 MB/sec IO throughput needed per core
- ✓ Allocate HBAs as necessary (minimum of 2 for redundancy) for total throughput
- ✓ Could be direct attached, fiber channel, ethernet (see below), or proprietary

Network

- ✓ Minimum of 2 x 10 gbit ethernet (one public, one private)
- ✓ Add more, as necessary, if storage is over the network (NAS, NFS, FCoE, etc)

Creating the Database

CREATING THE DATABASE

- Creation Options
  - Creation Options
    - Block Size and Page Size
      - Minimum should be 128K page size (8K block)
      - Use 256K page for larger databases (>500 GB or so where very large tables are involved)
      - Larger the page size the more RAM that will be required
    - CASE IGNORE vs. CASE RESPECT
      - Use CASE RESPECT whenever possible
      - Removes case comparison steps and improves performance
    - Java in the Database
      - Install if Java will be needed in the database
      - No impact if installed and not used

CREATING THE DATABASE

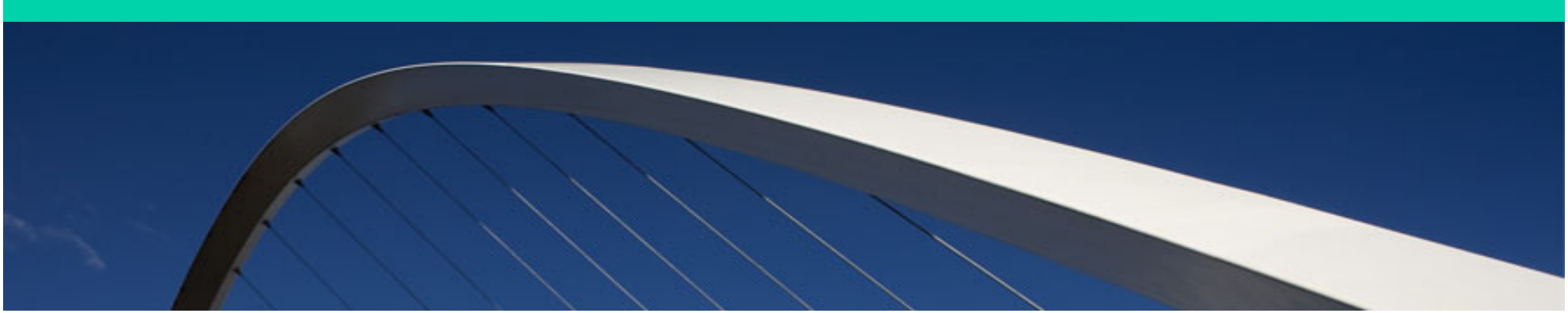
File Placement

- Use relative links – makes it easier to relocate files
- Use symbolic links for Sybase IQ MAIN and Sybase IQ TEMP
  - /IQM_devs/IQ_MAIN_00
  - /IQM_devs/IQ_TEMP_00
- Place the transaction log on significantly large file system
  - Recommend 10-20 GB of filesystem space per Sybase IQ node for database related storage
    - Transaction logs, Sybase IQ MSG, scripts, catalog file, etc.

CREATING THE DATABASE

File Locations

- Filename rules
  - Catalog: DATABASE_NAME.db
  - Transaction log: DATABASE_NAME.log
  - Sybase IQ MSG: DATABASE_NAME.iqmsg
  - Sybase IQ Main Store: DATABASE_NAME_iqmain_ooo
    - ooo would be an incrementing number
  - Sybase IQ Temp Store: DATABASE_NAME_iqtmp_ooo
    - ooo would be an incrementing number
  - Sybase IQ Configuration File: params.cfg (default)



SAP Sybase IQ Server Settings

SAP SYBASE IQ SERVER SETTINGS

- Query Performance
  - FORCE_NO_SCROLL_CURSORS
    - Should always be set to ON
    - Very few applications require this to be OFF
    - Can improve query performance

SAP SYBASE IQ SERVER SETTINGS

Server Side QUERY PLANS

Query Plan Settings to Provide Optimal Query Information to DBA's and Sybase Engineering

```
set temporary option query_plan='off';
set temporary option query_plan_as_html='on';
set temporary option query_plan_as_html_directory='qplans';
set temporary option query_plan_after_run='on';
set temporary option query_timing='on';
set temporary option query_detail='on';
set temporary option DML_options10='on';
set temporary option Query_Name = 'Query Name'
```

Should not be set globally as the Sybase IQ MSG file or query plan directory will grow rapidly

No need to set both Query_Plan and Query_Plan_As_HTML as plan output will be large

SYBASE IQ SERVER SETTINGS

Client Side QUERY PLANS

XML query plans can be viewed from DBISQL (Interactive SQL)

Can also retrieve HTML query plans after query execution and store on a local machine

New options to control remote query plans

QUERY_PLAN_TEXT_ACCESS = 'On' (default 'OFF')

Enables or prevents users from accessing query plans from the Interactive SQL (**dbisql**) client or from using SQL functions to get plans.

QUERY_PLAN_TEXT_CACHING = 'On' (default 'Off')

Allow users to specify whether or not Sybase IQ generates and caches IQ plans for queries executed by the user

SAP SYBASE IQ SERVER SETTINGS

Client Side QUERY PLANS

Retrieve an HTML query plan from the query plan cache to a local file:

```
select … from t1;
select HTML_plan();
output to c:\myplan.html
```

Retrieve an XML query plan from the query plan cache to a local file:

```
select … from t1;
select Graphical_plan();
output to c:\myplan.xml
```

SAP SYBASE IQ SERVER SETTINGS

- Storage
  - Append_Load (deprecated in IQ 16)
    - Can be used to improve load performance
    - Will not reuse Row ID's or the space occupied by those Row ID's
    - Great for systems where large, contiguous chunks of data are deleted
    - May not be good if random rows are deleted as it can lead to fragmentation and allocated, but unused space

SAP SYBASE IQ SERVER SETTINGS

Storage Continued

- Disk_Striping
  - If ON, Sybase IQ will stripe writes to all available devices
  - If OFF, the first device must be full before the next is used
  - Default is ON
- Disk_Striping_Packed
  - If ON, it forces better space usage and less fragmentation
  - Fragmentation is indicated when out of space messages are returned but the main dbspace used is less than 100% full
  - Allows Sybase IQ to better utilize small pieces of unused space that remain after compression
  - Default is ON

Connectivity

NETWORK CONNECTIVITY

TCP/IP Ports

- When configuring a TCP/IP port for Sybase IQ to listen on, use something other than the default of 2638
- SA and Sybase IQ use port 2638 as a broadcast listener
- Multiple servers on the same host share the default port for broadcast listening, but not for client network traffic
- Broadcast listening takes precedence over client network traffic as it is the first port started
- Easier and safer to assume that 2638 is in use already by SA/Sybase IQ and to use another port for client network traffic
- Many things must be considered if an Sybase IQ server is to use port 2638 for client network traffic on a host with more than one Sybase IQ server (server start order, multiplex synchronization, etc)
- See http://www.sybase.com/detail?id=1012955 for details

CLIENT ACCESS AND NETWORK CONNECTIVITY

There are different internal environments and settings for ODBC, JDBC, and Open Client connections

JDBC and ODBC are recommended

- JConnect is the Sybase ASE (Open Client) implementation of JDBC
- Recommend using the native JDBC driver "JDBC for iAnywhere"

Open Client

- Use with caution as most applications written using Open Client expect an ASE server and its T-SQL behavior
- Other than syntax differences, Open Client works great

CLIENT ACCESS AND NETWORK CONNECTIVITY

SAP IQ also includes and supports:

- Perl
- Python
- PHP
- Ruby

CLIENT ACCESS AND NETWORK CONNECTIVITY

OPEN CLIENT VS. ODBC

See Chapter 32 of the ASA Users Manual as well as the ASA Reference Section "Transact-SQL and SQL/92 compatibility options" for complete list of differences

If writing stored procedures or embedded application code, make sure to explicitly make settings for compatibility as these options will get set to different values for Open Client vs. ODBC connections

- ALLOW_NULLS_BY_DEFAULT
- QUOTED_IDENTIFIER
- STRING_RTRUNCATION
- ANSI_BLANKS
- ANSINULL
- CHAINDED
- FLOAT_AS_DOUBLE

CLIENT ACCESS AND NETWORK CONNECTIVITY

OPEN CLIENT VS. ODBC

- ODBC and JDBC connectivity support all Sybase IQ datatypes
- Older versions of Open Client may not support all Sybase IQ datatypes
  - Depending on version of Open Client, bigint, unsigned bigint, and unsigned integer will be OK (OC 12.5 and later)
  - Best to use most current Open Client versions shipped with Sybase IQ

CLIENT ACCESS AND NETWORK CONNECTIVITY

AutoPreCommit Within ODBC

- Set registry setting AutoPreCommit to Y
- Forces applications to issue a COMMIT before each query
- Go to the registry and update the corresponding Sybase Data Source Name (DSN) created, by adding a new value 'AutoPreCommit' with a value of 'Y'

  HKEY_LOCAL_MACHINE/SOFTWARE/ODBC/ODBC.INI/{DSN}

Packet Sizes

- Larger packet sizes will help with large data retrieval
- Use –p option in Sybase IQ configuration file to increase size
- Use CommBufferSize parameter in ODBC connection string

Maintenance Tasks

MAINTENANCE TASKS

DATABASE CONSISTENCY CHECKS

- Use the sp_iqcheckdb procedure to check the database consistency
  - Recommendations
    - Perform CHECK every 1 to 4 weeks
    - Perform VERIFY every 1 to 3 months
    - Perform REPAIR if errors are reported from VERIFY
  - Use RESOURCE_PERCENT parameter to control CPU usage during DBCC

MAINTENANCE TASKS

PARALLEL CREATE INDEX

Create Multiple Indexes in a batch

Syntax:

```
BEGIN PARALLEL Sybase IQ
    Create HG Index … ;
    Create LF Index … ;
END PARALLEL Sybase IQ ;
```

Recommendation

Create no more than 1 HG index per core and 2 other indexes per core on the host

Note: Two indexes on same column will be created in serial

MAINTENANCE TASKS

SAP SYBASE IQ MONITORING

- SAP Sybase IQ Monitor is a diagnostic tool for DBA's
- *sp_iqsysmon* is a great tool to consolidate the individual monitoring!!
- It collects and reports internal counters from the SAP Sybase IQ Buffer Caches
  - Main Cache interaction with the Sybase IQ Store
  - Temporary Cache interaction with the Sybase IQ Temp Store
- SAP Sybase IQ Monitor offers a series of "views" of the counters to showing differing aspects of the server and buffer cache workload

MAINTENANCE TASKS

SAP SYBASE IQ MONITORING

- Provides different views of buffer activity
  - summary report of both caches
  - detailed report of one cache
  - i/o activity of a cache
  - a debug report of all buffer cache activity
- You must specify an 'option' when you start the monitor to specify what view to monitor
- One Monitor may be running for a cache
  - May have one for each (Main & Temp)

MAINTENANCE TASKS

SAP SYBASE IQ OBJECT MONITORING

- May also use the newly released sp_iqsysmon procedure
- IQ also has an index advisor to recommend missing indexes
- IQ has tools to monitor object usage (workload manager)
  - **sp_iqworkmon, sp_iqtableuse, sp_iqcolumnuse, sp_iqindexuse**
- **Index advisor and workload manager information is cached in memory and will not persist through a restart**
  - **Suggest the events be written to safeguard information to persistent tables at regular intervals**

Data Model Recommendations

PROPER DATATYPE SIZING

Use the smallest datatypes possible for data

Be aware of all datatypes in Sybase IQ – there may be more than you know

If hour, minute and second information is not necessary, use DATE instead of DATETIME

If the data will fit within a TINYINT or SMALLINT datatype use that rather than INTEGER or BIGINT

Allows the engine to store data in smaller units (1-byte TINYINT or 2-byte SMALLINT versus 4-byte INTEGER or 8-byte BIGINT

Don't over allocate storage when defining NUMERIC() or DECIMAL() as it can be costly for data that doesn't need all that space

Default Column Storage (FP index) – IQ 15

Consider using Minimize_Storage option. This will place an SYBASE IQ UNIQUE(255) on every column for every table created and removes the need to use SYBASE IQ UNIQUE

If the value is <= 255 then Sybase IQ will place a 1-byte FP index on the column – 1 byte of storage per row

If the value is > 255 but <= 65536 then Sybase IQ will place a 2-byte FP index on the column – 2 bytes of storage per row

If the value is > 65536 and <= 16,777,216 then Sybase IQ will place a 3-byte FP index on the column – 3 bytes of storage per row (memory will effect the upper limit!)

May slightly hinder data loads, but improve query speeds

May incur onetime slight load slowdown while 2-byte FP is converted to flat FP, but this usually happens during the first load

SYBASE IQ UNIQUE and Minimize_Storage – IQ 15

Most customers now use Minimize_Storage option to allow IQ to chose the best FP index type on each column

- The FP index is the default storage mechanism

Can use the "IQ UNIQUE()" option to a column to force specific cardinality

- Used for FP index type (storage)
- Optimizer may make use of SYBASE IQ UNIQUE value at query runtime
- Does not need to be exact, but should be close to cardinality

Default Column Storage (n-bit index)– IQ 16

Introduced the n-bit index with IQ 16 as a replacement for the FastProjection index

On by default, so there is no need to set any options to take advantage of this

During an in-place migration from IQ 15 to IQ 16, columns will retain the older FP index

To take advantage of the n-bit compression and performance, run sp_iqrebuildindex on each column

SYBASE IQ UNIQUE and Minimize_Storage – IQ 16

IQ UNIQUE(0) disables all optimized default indexes

Avoid using IQ UNIQUE (0) for this reason

Minimize_Storage option does not apply to the new n-bit indexes

NULL VALUES

Always specify NULL or NOT NULL

- Open Client and ODBC connections have different default behavior when table is created

Allows the optimizer a better guess at join criteria

Will be compressed out when stored on disk

UNSIGNED DATATYPES

Use unsigned datatypes where possible

Use for surrogate keys and join columns

Unsigned data comparisons are quicker

LONG VARCHAR AND LONG VARBINARY

Can be used to store moderate amounts of text or binary data

VARCHAR() or VARBINARY() datatypes

- Maximum width is 32K (64K ascii hex for VARBINARY())

LONG BINARY

- Maximum width is 64K binary

The WORD index is the only index allowed on VARCHAR() data wider than 255 bytes

Storage will be allocated in 256 byte chunks

- A 257 byte string will require 512 bytes of storage
- Much less than the 2K requirement in ASE TEXT and IMAGE columns

LARGE OBJECT STORAGE

Can be used to store binary or text based objects

Extends the long binary datatype from a maximum size of 64K to an unlimited size

FP, WD, and TEXT indexes are the only viable indexes

Can be searched via the WD index now

Special function to return the size of an object (byte_length64)

Special function to return portions of the object, not the entire contents (byte_substr64)

VARCHAR VS. CHAR v15

- Use CHAR() whenever possible
- All storage in Sybase IQ is fixed width, so using a VARCHAR() over CHAR() will have minimal storage impact in IQ
- VARCHAR() types add slight storage overhead
  - A VARCHAR(100) columns will require 101 bytes of storage
    - 100 bytes for data
    - 1 byte for the size of data
- CHAR() data is blank padded, VARCHAR() is not
  - VARCHAR() is still stored as fixed width but any characters beyond the size are ignored
- If the average length of your data is significantly shorter than the column width, use VARCHAR in spite of the small storage overhead
  - LIKE predicate and most functional expressions usually have to search through large amounts of trailing blanks
  - More trailing blanks must be pushed through sort and hash operations

VARCHAR VS. CHAR v16

IQ 16 moved to variable width storage for varchar types

IQ 16 SP08 added statistics to varchar types to greatly help with query optimizations

Recommend using varchar in IQ 16 for this reason

WHEN AND WHERE TO USE INDEXES

Always use indexes on:

- Join columns (HG index regardless of cardinality)
- Searchable columns (HG index)
- Aggregation columns (HG)
  - Only if a single column is used in an aggregation
  - SUM(A * B) will not use an HNG, but rather an FP
- Date, Time, and Datetime columns (DATE, TIME, DTTM)

If uncertain, place an HG index on the column

Use Primary Key, Unique Constraint, or UNIQUE HG indexes where appropriate

WHEN AND WHERE TO USE INDEXES

A column with an CMP, WD, or any of the date indexes should have a corresponding HG index (very rare circumstances will negate this)

Indexes are not needed on columns whose data is ONLY returned to the client (projected)

Remove HNG indexes on date/time/datetime columns. Replace them with DATE, TIME, or DTTM indexes.

SIMPLE INDEX SELECTION CRITERIA – Main Indexes

Ask yourself these simple questions about each column to determine which indexes belong

Is the cardinality greater than 1500-2000?

Yes: place an HG index on this column

No: place an LF index on this column

Does this column contain date, time, datetime, or timestamp data?

Yes: place a DATE, TIME, or DTTM index on this column

SIMPLE INDEX SELECTION CRITERIA – Secondary Indexes

Will this column be used in range searches or aggregations?

Yes: place an HNG index on this column (if the aggregation contains more than just the column, an HNG may not be appropriate)

Does not apply to DATE, TIME, or DATETIME types

Will this column be used for word searching?

Yes: place a WD index on this column

Will this column be used for full text searching?

Yes: place a TEXT index on this column

Will this column be compared to another column in the same table?

Yes: place a CV index on the two columns being compared

Indexes to Avoid

- Investigate the use of HNG indexes. Over the past few releases quite a lot of the HNG features have been implemented in other indexes.
- **For this reason we do not recommend using HNG indexes**

- **Avoid using LF indexes from now on**
- The HG index has an identical structure for low cardinality data
- The HG index was overhauled to include all query optimizations of the LF index
- The HG index will load data into a low cardinality column with the same speed as an LF

MULTI-COLUMN INDEXES

The HG, UNIQUE HG, UNIQUE CONSTRAINT, and PRIMARY KEY indexes support multiple columns

GROUP Bys can take advantage of multi-column indexes so long as the index completely matches the column list and order

HG inserts are the most expensive in Sybase IQ with respect to other IQ indexes (IQ 15)

Much

Try to guarantee that inserts will happen at the end of the index

- Place generally incrementing data at the beginning of the index list (sequential data)
- For instance, a transaction date or batch number
- Something that will try to guarantee a sequential key

JOIN COLUMN

Prefer joining integer datatypes (unsigned if possible)

Integer comparisons are quicker than character comparisons

Keep the datatypes as narrow as possible to improve join performance by reducing disk I/O and memory requirements

Prefer using HG index on join columns rather than cardinality appropriate index (LF or HG)

PRIMARY KEYS

- Multi-column primary keys should have an additional LF or HG index placed on each individual column
  - Must be done manually
- UNIQUE CONSTRAINT, UNIQUE HG, and Primary Key are identical structures: HG index with no G-Array
- Use when possible
  - Helps optimizer make more informed query path decisions even if index is not used in joins or searches
  - You get an HG Index created automatically
  - This HG index has no G-Array (uses less space)

FOREIGN KEYS

Should be used to aid in query join performance

- Helps optimizer make more informed query path decisions even if index is not used in joins or searches
- You get an HG Index created automatically on the foreign key column in the current table
- Requires that a primary key exist on referenced table

Modeling for Scale Out

What is Scale Out?

Adding more cores to a single machine?

Adding more machines to the IQ Multiplex?

Adding more servers to the IQ logical server?

YES!

Indexes

Default Indexes should be enough in a lot of situations

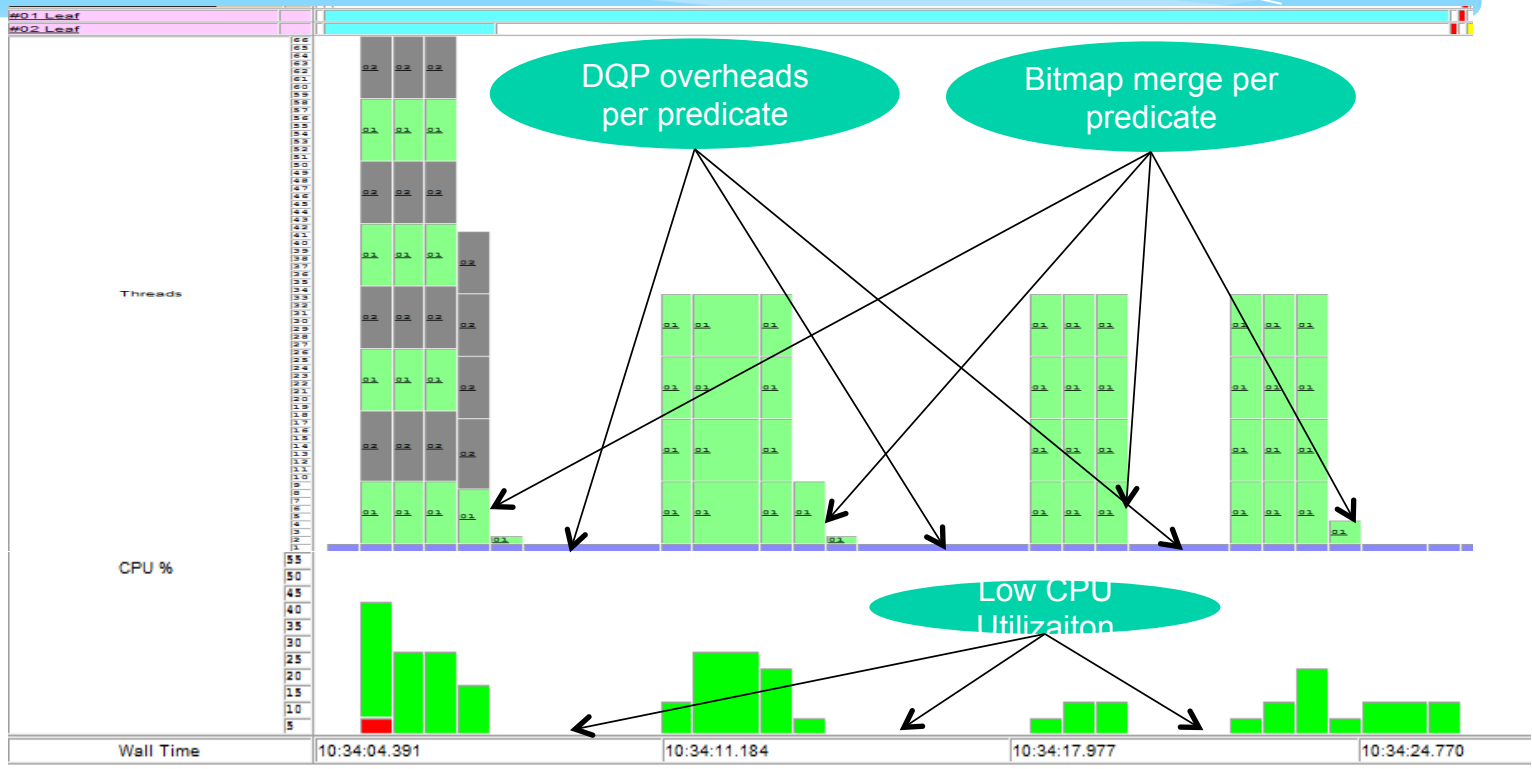
Allows the optimizer to choose the best, fully parallel, execution path

Allows the optimizer to force DQP/PlexQ

The n-bit indexes contain a lot of statistics that didn't exist before and are lowering our reliance on other indexes (HG, CMP, etc)

Predicate Scale out Problem

- Each predicate is a separate DQP fragment
- Query has 4 predicate fragments on table
- DQP overheads per predicate
- Serialize/De-serialize fragment and parallel execution setup
- Bitmap Merge per predicate
- Merge bitmaps from parallel workers to single result bitmap



TPCH Q19 scale 1000 – Predicate execution (15.x)

Predicate scale out Solution

Combine Predicates

Predicates with same semantic work partitioning combined into a single fragment

Example

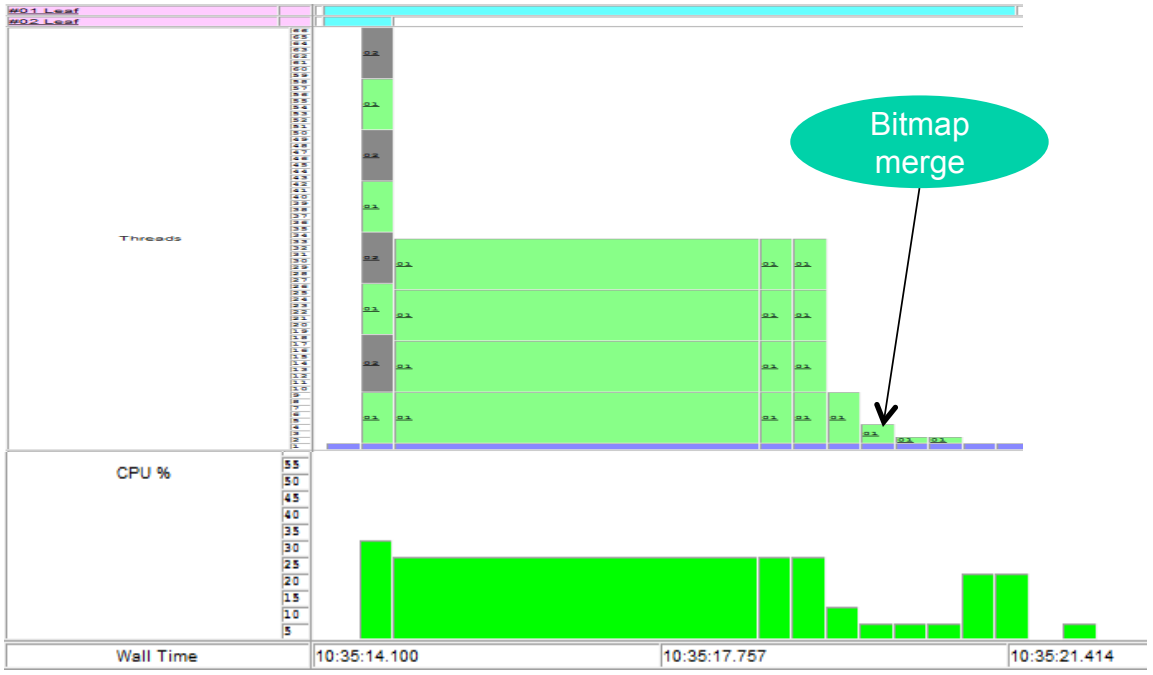
All predicates using row range partitioning, combined to form one single predicate fragment

Result

No DQP overheads per predicate

One time bitmap merges at the end of combined fragment

Better CPU utilization



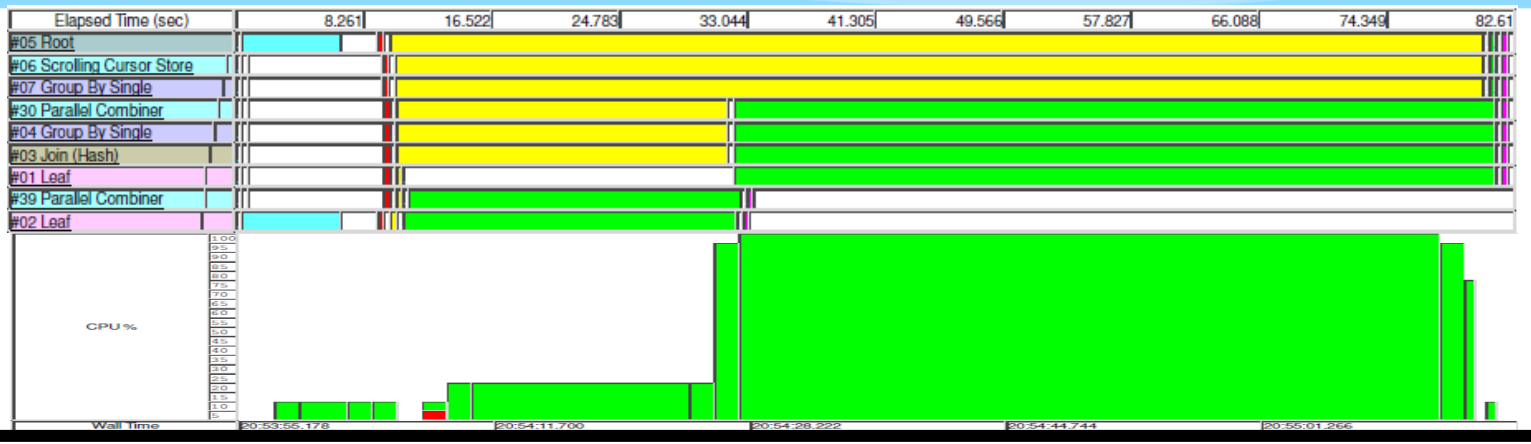
Hash Partitioning

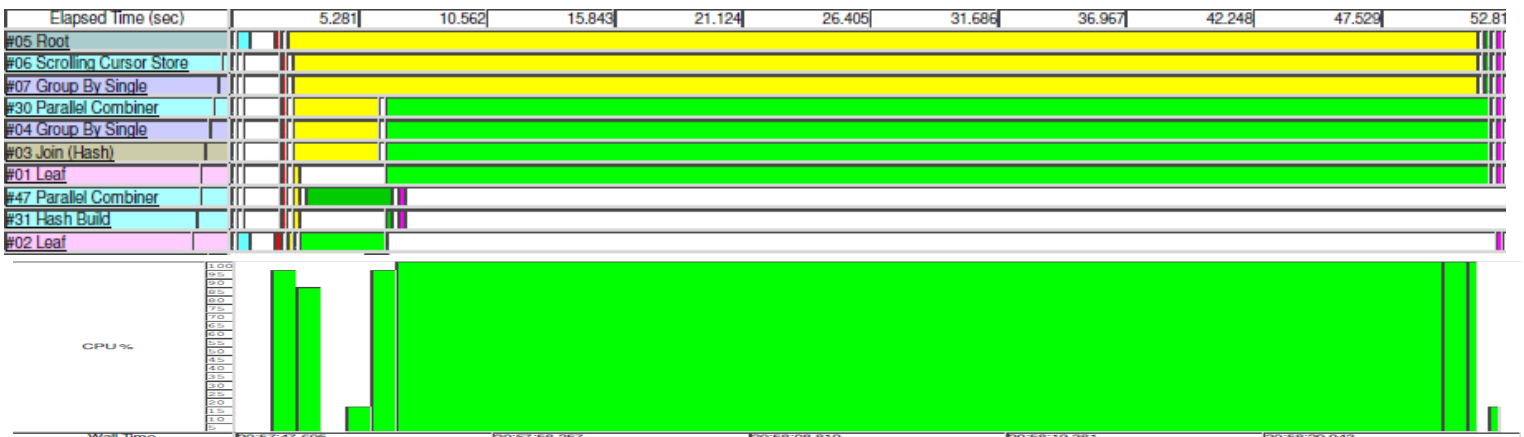
Important for DQP

Helps the optimizer track where data is (node affinity)

Helps the optimizer better break up the workload for queries and distribute the work unit (partition) on different machines

Parallel Hash Build –timing changes

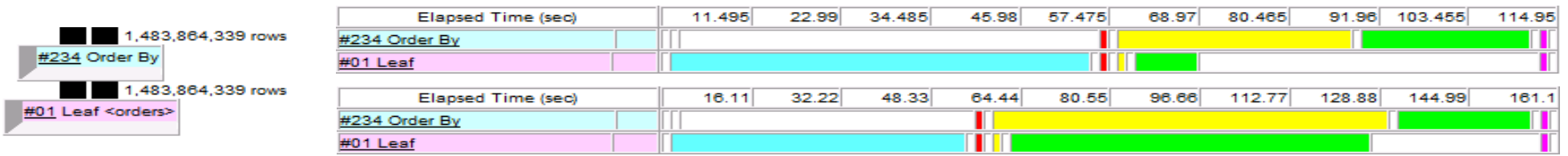


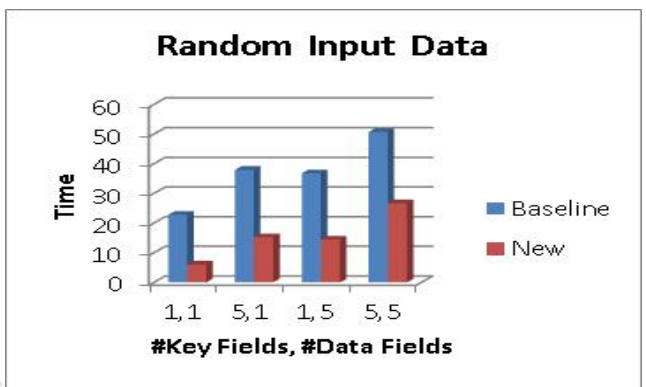


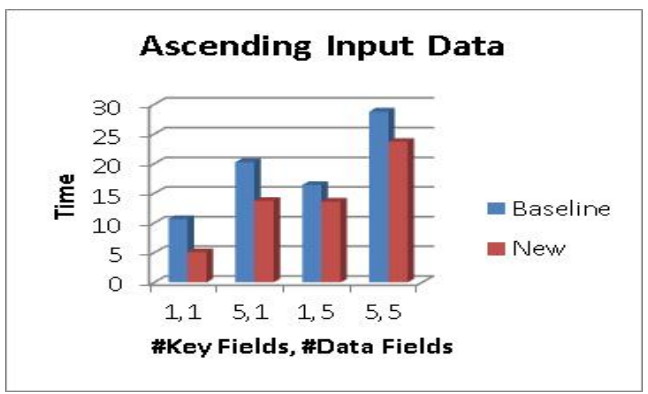
Sorting Enhancements

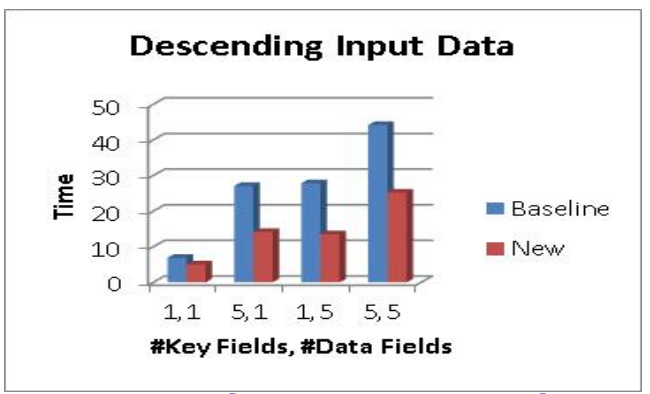
Sorting scales better with number of cores

Temp usage reduced for narrow data









DQP Tuning Mechanisms

DQP Tuning
System-level

- Caches
  - Use large temp cache to maximize hash processing
- Disks
  - Ensure reasonable bandwidth for all cores across multiplex
  - Move bandwidth between main and shared temp to address IO bottlenecks
- Schema
  - Use minimize_storage or iq_unique to produce 1/2/3-byte storage (IQ 15 only, leave at default to leverage n-bit indexes in IQ 16)
  - Add HG indexes where maintenance costs are manageable
  - Avoid normalization that requires large intermediate joins
  - Use partitioning to collocate data and avoid sparse projection

DQP Tuning

Joins

- Avoid merge join sorting
  - Small join ratio - Reduce dml_options48 to prefer hash joins
  - Use HG indexes to leverage ordered leafs with natural semijoins
- Option Max_Hash_Rows
  - Set just high enough to eliminate unwanted merge joins
- Join Result Constraint
  - Add appropriate constraints or indexes for optimizer to determine join relationships
  - Check that many-many joins are really that

DQP Tuning
Group By

- Options that affect group count estimates
  - Groupby_Column_Correlation (15.4 ESD1)
  - Groupby_Table_Correlation (15.4 ESD1)
  - Dml_options75 – exact group count (15.1ESD3)
- Hash → Sort overflow
  - Undersized estimates may overflow hash quota
  - Performance may still be good compared to GROUP BY sort
- Hash_Pinnable_Cache_Percent option
  - Can be safe to increase if you also limit joins with Max_Hash_Rows

DQP Tuning

DQP Runtime Sizing

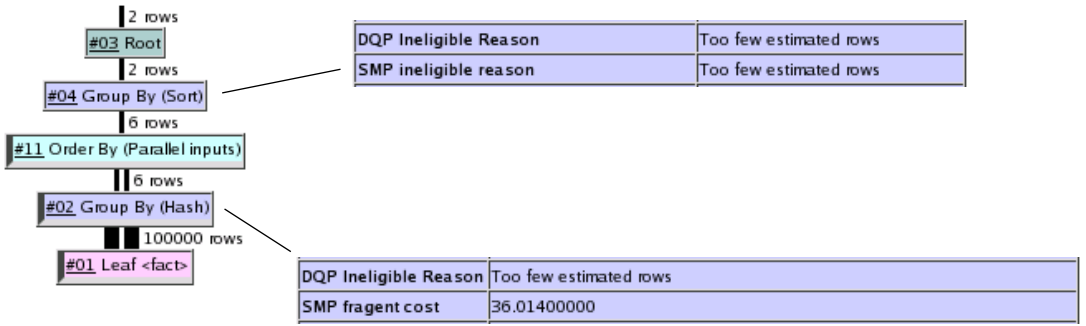
Work Unit Sizing

- Options Dml_options51 and DML_Options52 affect initial work unit size
- Decrease work unit size to increase parallelism and distribution

Work Units - sample_nc110688	194 (24, 24, 26, 24, 24, 25, 24, 23)
Work Units - nw24909_sample	222 (28, 29, 28, 28, 28, 27, 27, 27)
Max. Possible Parallel Arms	16
Max. Active Parallel Threads	16
Parallel Sink Work Units	416
First Worker Work Unit	9
Max. Possible Parallel Arms	8
Parallel Source Work Units	120
Initial Source Work Unit size	3923461
Optimization Note	Primary Parallel Join Source

Fragment Cost

- May limit Max Possible Parallel Arms or not distribute if too few work units
- Ineligible Reason shown when not enough work units for SMP or DQP cores



Data Loading

Load Table

Parallel Load Table (server side loads only)

- Make sure that the data file has a column delimiter after last column just before the row delimiter
- Must use ROW DELIMITED BY and DELIMITED BY options in load table command (for FORMAT ASCII and FORMAT BCP)
- FORMAT ASCII and FORMAT BCP want to run parallel
- Column and row delimiters should be a single character
- Binary data files are always done in parallel
- Can also be downgraded to a serial load (flat file reading only) if
  - Max_IQ_Threads_Per_Connection and Max_IQ_Threads_Per_Team, and/or -iqmt are too low
  - Not all columns in the table are being loaded (partial width insert)

Data Loading Notes

On Linux, SAP IQ can read gzip compressed files in native format! Performance will suffer, however, as decompressing the file is single threaded.

Client side loads are still mostly single threaded

IQ 16 can load 10-20 MB/sec per core of raw data using server side loads

- A 40 core machine can load 400-800 MB/sec of source ASCII data
- Assumes the hardware can serve it that fast

Load Table

Can use FILLER() clause with a delimiter or byte count

Better performance achieved by casting the date or datetime formats rather than letting Sybase IQ guess

If possible, issue a single load table with multiple files rather than 1 load table per file to be loaded into a table as each file will be processed in parallel

Insert… Location

Easy way to move data from any Open Client source (ASE, ASA, IQ, Direct Connect gateway to non-Sybase RDBMS sources)

Syntax:

```
insert into TABLE()
location 'SERVERNAME.DBNAME'
{ select statement };
```

Sybase IQ username and login must match on remote system

Interfaces entry must match the SERVERNAME

Can also be used to move data quickly from an ASA table to an Sybase IQ in the same server

Moving text/image (blob/clob) data has a limit that is version specific (so check the manuals for the per version limits)

Single Row Operations

Avoid at all costs for large data manipulation operations

Different from single statement operations that modify many rows

Individual INSERT … VALUES() will be slower than bulk load operations

- Expect no more than 5,000 to 20,000 operations per hour

Single Row Operations

If multiple single row inserts are needed there is a method to improve performance

Consider this code (1 insert per row):

```
insert into #my_table values ( 1, 1 )
insert into #my_table values ( 2, 1 )
insert into #my_table values ( 3, 1 )
```

Can be converted to this (only 2 inserts):

```
create table #iq_dummy ( a1 int )
insert into #iq_dummy values ( 1 )
insert into #my_table select 1, 1 from #iq_dummy
union all  select 2, 1 from #iq_dummy
union all  select 3, 1 from #iq_dummy
```

Named Pipes

Named pipes can be faster – eliminate disk I/O

How to make a named pipe on most flavors of UNIX:

```
mknod PIPE_NAME p
```

Can use BCP, GZIP, UNCOMPRESS, or applications to push IQ formatted data into named pipe

LOAD TABLE command can read from named pipe

Can also fast extract data to named pipes so that they can be read by another application or even compressed and stored

Investigate OS tuning to increase the size of a named pipe as the buffer size is typically small at 8k

Partial Width Inserts (pre-v16)

Can induce fragmentation and overly large space consumption if not watched

Row ID's from deleted data are not reused during a partial width insert operation

Space from delete data is not reused (because Row ID's are not reused)

Partial width inserts are analogous to APPEND_LOAD='on' in terms of Row ID and space behavior

Only becomes a problem if partial width inserts are a way of life for a table

No longer possible in IQ v16 and later

Notes

Tuning is only as good as the CPUs, RAM, storage, SAN, etc

See my IQ Hardware Sizing Guide or the presentation on IQ Sizing for more details

Other Resources

Contact me via email at mark.mumy@sap.com

Check out my blog: http://scn.sap.com/people/markmumy/blog

Use the IQ Community on the SAP Community Site:
http://scn.sap.com/community/sybase-iq

Use the IQ Users Group: iqug@iqug.org

Questions and Answers

Thank You for Attending

Please complete your session feedback form