The Four Types of Indexes

If I were to ask you how many types of indexes existed for DB2 for i5/OS what would your answer be?

The obvious answer for those of you knowledgeable about DB2 would be two – binary radix and encoded vector indexes. Binary radix indexes are used by logical files and SQL Index objects and have been around for over 20 years. The second flavor, encoded vector indexes (EVIs), was introduced in the late 1990's to help speed up decision support queries.

So how can there be **four** types of indexes? Well, I really asked a trick question! For the purposes of this article the four types of indexes are:

- ✦ Boat anchors – the indexes that are not really used or are not necessary and just taking up space and resources on your system.
- ✦ Sub-optimal – these indexes are useful, but can be optimized in some way to make them more efficient or better for queries or RPG programs.
- ✦ Optimal – these indexes are also useful and are as efficient as they can be.

- ✦ Needed – these indexes **don't exist** in your database but should so that queries perform better.

Centerfield's products have been helping determine which new indexes are needed for over eight years – and continue to improve by leveraging new opportunities offered by the SQL query engine in V5R3 and V5R4. In our upcoming release we now assist you with the logical files and SQL indexes that **currently** exist in your database but that can be optimized.

Boat anchor indexes

The "boat anchor" category includes logical files and SQL indexes that truly are not used or are duplicates of another logical file or index. To determine whether a logical file or SQL index is truly unused is not as simple as looking at the output of the DSPFD command. An index may actually be used by the query optimizer and be necessary even if it appears to be unused. Centerfield's product takes this fact into

UNUSED   NOT OPTIMAL   OPTIMAL   DO NOT EXIST BUT NEEDED

Physical File or SQL Table
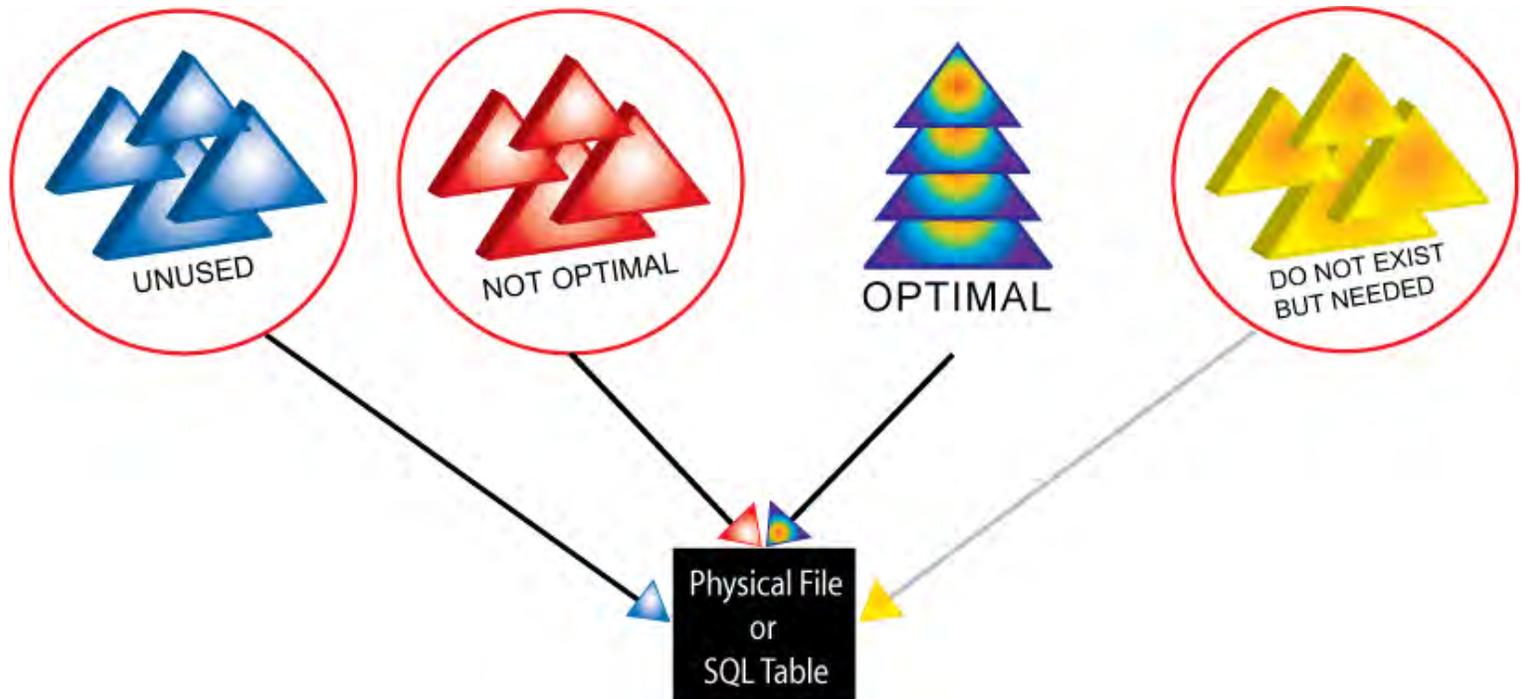
Getting data out of other job's

QTEMP

For as long as developers have been using the QTEMP library to hold temporary work files and such, there has been a need to have access to them for debugging purposes. IBM teases us by letting you see other job's library list via commands like:

- DSPJOB JOB(160206/OTHERUSER/OTHERJOB) OPTION(*LIBL)
- option 5 against QTEMP

and letting you see the contents of QTEMP, but not letting you access data in those work files, user spaces etc.
If the developer is around, he can instrument the program to put temporary work files in a non-QTEMP library and debug it in that fashion. If developer is not around (i.e. 3rd party application) and you are system administrator, you'd think you're stuck. Well, not so.

I've had this same need while working on some exit point programs and figured there has to be a way to get to QTEMP's data. I mean, I have QSECOFR authority on this system, I should be able to do whatever I want!
Lo and behold, fellow named David Ong has written just such a utility and was generous enough to share it at Search400 tips website:

http://search400.techtarget.com/tip/1,289483,sid3_gci816185,00.html

I've used it great number of times without giving it a second thought and proved it invaluable. Alas, when we upgraded to V5R4 , this technique stopped working due to some low level operating system changes IBM instituted. I still needed access to this nice utility and when I read up more on IBM changes in V5R4, I realized I could use new Job Interrupt support to accomplish the same thing.

2

DB2 Expert on a SILVER Platform

Historically, iSeries shops have had little need for a true database administrator. Most of the complex work required of database administrators on other platforms is handled automatically by DB2 for i5/OS. Good examples include disk storage and memory buffer allocation. Other databases require the administrator to configure and monitor the use of these resources while the iSeries requires neither. Because of this, routine database tasks are normally handled by a programmer or a skilled system administrator.

In the last five years however, the **rules have changed**. Proactive, focused management of DB2 has become a critical system function because of:

✦ Complex, distributed ERP applications which use the System i as a database server

✦ The unbelievable growth of historical data kept in DB2

✦ A large number of unpredictable queries generated by end-users

✦ Web applications that must be available 7x24 and perform well to keep customers and business partners happy Data mart or data warehouse initiatives

The business drivers behind the heavy use of DB2 mean that shops are devoting more and more time to database issues.

The Road to DB2 expertise

So can a System i shop obtain DB2 expertise?

The first option is obviously to hire an expert. Unfortunately, people with deep skills are very difficult to find. The second option is to assign someone to the task. If a programmer was assigned to be a DBA, what are the necessary steps for that person to become a true DB2 expert?

Step 1: Education

To become an expert in DB2 for i5/OS, the first step might be to delve into IBM's documentation. Surprisingly, over 11,000 pages of DB2 manuals exist. This number does not even include programming, security, or backup and recovery information.

A second step would be to take education classes that are available from a variety of sources including IBM and COMMON. Additionally, on-line webinars can be attended for very little cost and provide targeted information about DB2.

Step 2: Identify best practices

The next step would be to take information from the IBM manuals or other sources and identify the "best practices" to manage DB2. These best practices, if applied correctly, translate into better performance, more secure data, and higher availability. Centerfield has identified several hundred of these best practices that are applicable in a variety of situations that we surface in our products, support, consulting, and training.

Step 3: Analysis

The third step is to examine production databases and identify opportunities to apply best practices. Unfortunately, most production databases have between 2,000 and 10,000 objects to examine so this becomes a very challenging task.

Step 4: Prioritize actions

After analyzing the database and identifying useful changes, the next step is to prioritize those activities. Clearly the most beneficial items should be prioritized first to get the most immediate value with the least amount of work.

Step 5: Implement actions

Once the high-value changes are identified, the next step is to implement those changes. This can be as simple as running a single command or as complex as writing a long program. A common oversight at this point is supplying some way to back out a change. If something does not work in a production environment, it isn't going to look good if there wasn't a back out plan put in place.

Step 6: Feedback

Any time changes are made to improve a database, it would be useful for many reasons to get feedback. While the expected outcome of any change is positive, there are situations where it isn't always possible to tell beforehand what **actual** benefit would result. Therefore, a feedback loop is necessary to verify changes were positive. Another benefit of having feedback is that good news can be delivered to peers and management about the changes.

A second road – Centerfield Technology

Centerfield has provided DB2 tools for administrators since 1998. Over that time, these tools have given administrators a way to help tune DB2 and applications without needing to become an expert.

As I hinted at in our last newsletter, we are working feverishly on the final touches of HomeRun Version 6.0. Our objective is to make the new version generally available in May.

Unlike a typical software upgrade, this new version of software represents a major step forward in the management of DB2 for i5/OS and in the way our products are used. HomeRun 6.0 has effectively redefined what it means to manage SQL indexes and logical files on the iSeries and System i. For more details about these new index management components, see the "The Four Types of Indexes" article in this newsletter.

What is even more exciting than the new index management functions, is a new component we've dubbed Autonomic Database Assistant (AutoDBA). This key piece of software provides a revolutionary way to bring expert database advice to shops small and large. AutoDBA provides advanced functions to analyze database objects and recommend "best practices" to system administrators and development personnel – without the need to read through over 11,000 pages of DB2 documentation. What this means in practical

terms is that our customers will now be able to get expert advice with very little effort – and a safe, secure, and auditable way to implement that advice.

Centerfield's long-term plan is to use the AutoDBA platform to deliver expert advice to improve a **critical** part of your application stack – your DB2 database. For more information on AutoDBA see the "DB2 Expert on a Silver Platter" article.

The team here at Centerfield hopes you enjoy this newsletter. As always, feel free to give use feedback and suggestions on how we can make it even better!

Best Regards,

Mark L. Holm
Chief Technology Officer
Centerfield Technology, Inc.

---

account as it analyzes the database and makes recommendations.

Dealing with these unused or redundant indexes has always been tricky for system administrators. On one hand they know that these unnecessary objects introduce overhead to their applications, but they don't want to take the risk of deleting an object that *might* be needed in the future. Centerfield's HomeRun product has solved this problem by introducing advice and a methodology for reducing the impact of these objects while removing the risk to production applications.

Sub-optimal indexes

Sub-optimal indexes come in a wide variety of flavors. Of those, let's review the three types Centerfield's tools optimize well:

Select/Omit logical files. One of the most common situations is the over-use of select/omit logical files. In the past five years we've analyzed a large number of customer databases. It is very common to find 10-20% of existing select/omit logical files that contain over 90% of the data in the file. Ideally, these files would share an existing logical file and use the DYNSLT keyword. This essentially reduces the number of indexes associated with a physical file and improves batch elapsed time, reduces memory use, and makes backups faster.

High-use indexes. Another opportunity for many RPG shops is to optimize their most-used logical files. Often, logical files are processed in sequential order which results in a high number of small, random disk I/Os. These I/Os result in slow response

times or long running batch processes. It is common for an iSeries customer who has upgraded to a much faster system to find batch work not getting done any faster than it did before the upgrade. The root cause of this can often be traced back to disk I/O that could be significantly reduced and allow the batch job to take advantage of the much faster processors.

Clustering candidates. Clustered indexes, a technology that has existed for other database systems for years, are another way to optimize a database without the need for application changes. Essentially the idea is to rearrange the order of the data in the file or table to match the order of the most popular index (where popularity means the number of rows accessed in sequential order). The latest releases of DB2 for i5/OS do not support clustering directly, but do have all of the necessary function to allow database administrators to gain the advantages of the support. The trick is to identify **which** index is the best candidate for clustering and **implementing a strategy** to get the rows in the right order.

The good news is that the soon-to-be-available release of HomeRun (Version 6.0) contains functions that allow system administrators, programmers, or database administrators to easily identify and manage the four types of indexes – without adding to their already busy workloads. Most importantly, it provides the capability to implement changes in a safe and secure manner while gaining the benefits of the expert advice. Finally, the product provides feedback so management can be easily informed of the positive difference the changes have made on their important applications database.

With HomeRun 6.0 you can now go from the four types of indexes to two – efficient binary radix and efficient encoded vector indexes.

I got to programming and got it to work, so I'll share it with you fine readers in case you have a need to do the same.

Step 1 – compile the executor program (to be called by the system)

```
/********************************************************************************/
/* Elvis Budimlic created on 10/08/2006.                             */
/*                                                                   */
/* MYEXECPGM - Job Interrupt exit point program.                     */
/*          Executes arbitrary command initiated from another job via */
/*          RUNJOBCMD.                                               */
/*                                                                   */
/* Must do couple of things to enable RUNJOBCMD to work:             */
/*   - CHGSYSVAL SYSVAL(QALWJOBITP) VALUE('2')                        */
/*   - ADDEXITPGM EXITPNT(QIBM_QWC_JOBITPPGM) FORMAT(JITP0100) PGMNBR(*LOW) */
/*                PGM(QGPL/MYEXECPGM) THDSAFE(*YES) MLTTHDACN(*RUN)   */
/*                TEXT('Job Interrupt executor')                      */
/*                                                                   */
/* To compile:                                                       */
/*     CRTBNDCL PGM(QGPL/MYEXECPGM) SRCFILE(QGPL/QCLSRC) SRCMBR(MYEXECPGM)   */
/*                                                                   */
/********************************************************************************/
pgm (&command &len)
dcl &command *char 300 /* could be 2000, but for IBM bug (fixed in next OS?) */
call qcmdexc parm(&command 0000000300.00000)
endpgm
```

Step 2 – compile the Command processing program for RUNJOBCMD

```
/********************************************************************************/
/* Elvis Budimlic created on 10/08/2006.                             */
/*                                                                   */
/* JOBINTCPP - command processing program for RUNJOBCMD command      */
/*                                                                   */
/* Must do couple of things to enable RUNJOBCMD to work:             */
/*   - CHGSYSVAL SYSVAL(QALWJOBITP) VALUE('2')                        */
/*   - ADDEXITPGM EXITPNT(QIBM_QWC_JOBITPPGM) FORMAT(JITP0100) PGMNBR(*LOW) */
/*                PGM(QGPL/MYEXECPGM) THDSAFE(*YES) MLTTHDACN(*RUN)   */
/*                TEXT('Job Interrupt executor')                      */
/*                                                                   */
/* To compile:                                                       */
/*     CRTBNDCL PGM(QGPL/JOBINTCPP) SRCFILE(QGPL/QCLSRC) SRCMBR(JOBINTCPP)   */
/*                                                                   */
/********************************************************************************/
pgm (&inJobName &inJobUser &inJobNum &inPgmData)

dcl &inJobName    *char   10
dcl &inJobUser    *char   10
dcl &inJobNum     *char    6
dcl &inPgmData    *char  300 /* could be 2000, but for IBM bug */
dcl &input        *char 2056
dcl   &pgmName    *char   10 stg(*defined) defvar(&input  1)
dcl   &pgmLib     *char   10 stg(*defined) defvar(&input 11)
dcl   &jobName    *char   10 stg(*defined) defvar(&input 21)
dcl   &jobUser    *char   10 stg(*defined) defvar(&input 31)
dcl   &jobNumber  *char    6 stg(*defined) defvar(&input 41)
dcl   &reserved   *char    2 stg(*defined) defvar(&input 47)
dcl   &dataOffset *int       stg(*defined) defvar(&input 49)
dcl   &dataLength *int       stg(*defined) defvar(&input 53)
dcl   &pgmData    *char 2000 stg(*defined) defvar(&input 57)

chgvar &pgmName    'MYEXECPGM '
chgvar &pgmLib     'QGPL      '
chgvar &jobName    &inJobName
chgvar &jobUser    &inJobUser
chgvar &jobNumber  &inJobNum
```

(Getting data out of other job's QTEMP — Continued from page 5)

```
chgvar &reserved   x'0000'
chgvar &dataOffset 56
chgvar &dataLength 300      /* could be 2000, but for IBM bug */
chgvar &pgmData    &inPgmData

call QWCJBITP (&input 'JITP0100' x'0000000000000000')

endpgm
```

Step 3 – compile RUNJOBCMD command you'll use when you want to access other job's QTEMP

```
/*****************************************************************************/
/* Elvis Budimlic created on 10/08/2006.                                   */
/*                                                                         */
/* RUNJOBCMD - runs arbitrary command in another job                       */
/*                                                                         */
/* Must do couple of things to enable RUNJOBCMD to work:                   */
/*    - CHGSYSVAL SYSVAL(QALWJOBITP) VALUE('2')                            */
/*    - ADDEXITPGM EXITPNT(QIBM_QWC_JOBITPPGM) FORMAT(JITP0100) PGMNBR(*LOW) */
/*              PGM(QGPL/MYEXECPGM) THDSAFE(*YES) MLTTHDACN(*RUN)          */
/*              TEXT('Job Interrupt executor')                            */
/*                                                                         */
/* To compile:                                                             */
/*     CRTCMD CMD(QGPL/RUNJOBCMD) PGM(QGPL/JOBINTCPP) SRCFILE(QGPL/QCMDSRC) */
/*             THDSAFE(*YES)                                               */
/*****************************************************************************/

               CMD        PROMPT('Run job command')

               PARM       KWD(JOBNAME) TYPE(*CHAR) LEN(10) MIN(1) +
                            CHOICE('Name') PROMPT('Job name')
               PARM       KWD(JOBUSER) TYPE(*CHAR) LEN(10) MIN(1) +
                            CHOICE('Name') PROMPT('Job user')
               PARM       KWD(JOBNUMBER) TYPE(*CHAR) LEN(6) +
                            RANGE('000000' '999999') MIN(1) +
                            CHOICE('000000 - 999999') PROMPT('Job +
                            number')
               /* length could be 2000 but for IBM bug */
               PARM       KWD(COMMAND) TYPE(*CHAR) LEN(300) MIN(1) +
                            CHOICE('Name') PROMPT('Command to run')
```

Step 4 – register the exit point as illustrated in the comments section of the programs above (I duplicated the instructions in all comment headers intentionally)

Ok, so now you have all things in place and want to pull data from another job's QTEMP library you could do something like this:

RUNJOBCMD JOBNAME(OTHERJOB) JOBUSER(OTHERUSER) JOBNUMBER(160206)
COMMAND('crtdupobj workfile qtemp *file qgpl data(*yes)')

The example above duplicates "workfile" to QGPL library so now you can check what data is in it and verify why that job is not doing what it's supposed to.

The potential uses (and abuses) of this methodology are endless. Thus, make sure you get approval from your security officer before putting something like this in place. Hope this helps you in those tricky situation where 3rd party support says, "if we just knew what data is in that file…".

Happy reading!

Elvis

IBM and mySQL

AB just announced a partnership offering which created some buzz in the System i world. Anything that deals with DB2 on System i is our business, so I decided to investigate and weigh in with what I've found so far. Keep in mind these are **my** thoughts on the announcement and represent things as seen from my perspective.

Let's dig in

We've all heard of the open source database called mySQL. It is very popular download on the Web claiming to have 11 million product installations. So what and who is mySQL AB? In Sweden, AB stands for Aktiebolag meaning 'corporation offering public stock'. As you might suspect, this company has Swedish roots and actually is dual headquartered in Upssala, Sweden and Cupertino, California. The company draws revenue from consulting and a not-so-open-source variation of the GPL license.

The primary attraction of mySQL is its open-source GPL license version that is freely downloadable and easily adaptable. These two factors (free & easy) have made it very popular with scores of developers out there who have built on top of it and provided great number of popular applications (Google uses mySQL).

You are aware of costs associated with purchasing database software for Oracle or SQL Server. Alternatively, DB2 on the System i is embedded into the i5/OS but as we all know that is one of the contributing factors of the high up-front costs of purchasing an i5/OS license. Now imagine if you're a lonely developer starting to develop a new web application that requires database access and have to cough up the cost of purchasing a database for your development. It's not going to be DB2 for System i since you'll also have to purchase the hardware. It could be DB2 for LUW or SQL server or, you guessed it, free copy of mySQL.

After an application is developed it'll probably end up being distributed to your customers. If it's developed with mySQL being a target database, chances are that'll be preferable database choice for distribution to customers as well as it's simply easier to support what you already know.

This scenario is so common that there is actually an entire new programming paradigm out there with huge following called LAMP. LAMP is solution stack of free software programs used together to run dynamic Web sites or servers – Linux, Apache, MySQL, PHP. You can develop a full fledged Web application and not pay a dime!

So far, mySQL looks like a winner. But being that our specific expertise is database, I am a bit skeptical about **fairytales** like that, since the amount of work required for a world class database is truly immense. Storage, journaling, security, row level locking, backup, error recovery, transaction isolation, performance tuning, national language support, hardware support…. list doesn't end there. I don't care how good open-source developers are, they can't do it all!

Thing called storage engine

As I read deeper into the announcement and key words jumped out me and that is that System i will serve as a mySQL **storage engine**. So let's take a deeper look at what that really means.

Suffice it to say, the storage engine does the dirty work! The hard work includes memory, index and storage management as well as journaling, concurrency, transaction support, referential integrity, recovery and physical disk interfacing. mySQL Server takes care of the programming language interface, connectivity, connection pooling, DDL, DML, access paths, caching, threading and like. I realize this is gross oversimplification but I can't cover the entire architecture in this article nor do I want to.

When I learn about something new (e.g. storage engine concept), I love to see how it stacks up against something I already know or alternatively against its competition. In this case I found a comparison of different storage engines that support mySQL and after reading into it and other descriptions, a light came on for me. Though there are many common elements across different storage engines, each engine excels at some particular thing: retrieving data fast, retrieving large amount of archived data, data warehousing etc. Each storage engine seems to have a niche that makes it relevant in mySQL arena.
There are one or two storage engines that could pass as transaction oriented storage engines. One that stood out for me is InnoDB and is (importantly) owned by Oracle.

Guess which system with integrated database **excels** at OLTP – our good old System i! This really piqued my curiosity as pieces started to fall together. For those enterprises that run high number of transactions mySQL->System i combination may make perfect sense. ALL of the businesses I talk to use the System i for OLTP and have been doing so for 20-30 years. What these businesses complain about is number of available applications for the System i platform.

This combination doesn't necessarily fit the description I had earlier of the lonely developer using free software for development, but it may fit commercial ISVs that are looking for cost effective total enterprise solution (and for some reason don't want to use DB2.

Again, I know I am greatly oversimplifying things here and that both mySQL and System i can and

With our latest version of our HomeRun product, we have created a solution that allows System i shops to perform the six steps described above – **with their current IT personnel**. The bottom line is that it removes a key reason that new initiatives are not being implemented; especially those that require heavy use of DB2 like ERP upgrades, new web applications, and data warehousing.

We are introducing a component of HomeRun called Autonomic Database Assistant (AutoDBA for short) that provides expert DB2 advice (i.e. best practices) and allows the process to be almost fully automated. In the first release of AutoDBA, automated index

management is the focus. Proper management of SQL indexes and logical files is a very important part of DB2 administration and almost all System i customers will see value from the advice provided by the product.

Unlike most administration products, the purpose behind AutoDBA is not to make it easier to analyze data – the purpose is to provide answers without having to take the time to analyze data. Many tools provide huge amounts of data and many even display the data in visually appealing graphics, but leave it up to you to draw conclusions.

So let's look at the above steps required to become a DB2 Expert with Centerfield's HomeRun product:

Step 1: Education – very little is

necessary; AutoDBA has done the learning for you.

Step 2: Identify best practices – Not necessary; AutoDBA knows the best practices

Step 3: Analysis – Not necessary, AutoDBA does the analysis

Step 4: Prioritize actions – Not necessary; AutoDBA prioritizes its advice

Step 5: Implement actions – Some involvement required; AutoDBA can automatically implement some actions, others require manual action. All automated actions can be easily backed out.

Step 6: Feedback – Not necessary; AutoDBA provides feedback on actions taken.

---

are doing great number of other things. But if the System i is going to compete with ten other supported mySQL storage engines and an unknown number of community supported ones, we need to be realistic and recognize where System i can win hands down and where it cannot. OLTP is where I see this being the case and purchasing over web is a specific example.

An indirect benefit of this setup is that now a development house can choose a different storage engine for the type of work where System i doesn't excel (i.e. Memory storage engine which stores all data in RAM for extremely fast access in environments that require quick look ups of reference and other like data) and leverage best of both (or multiple) worlds.

How are they doing it

The techie in me wants to know how the System i would be used as storage engine. Of course, the solution is not native in a sense that it won't run directly on System i as your RPG, COBOL and C programs do. Remember, we have DB2 for that.

You will be using customized mySQL for AIX/POWER package to run as a binary for i5/OS in **PASE** (Portable Application Solutions Environment). The best piece of news is that the DB2 storage engine will be a layer of code developed and tested by IBM. This piece will then be distributed by mySQL as part of customized package I mentioned.

What I like about this arrangement is that IBM is taking on work that it can do best and leaving rest of the work (packaging) to mySQL, which does that piece very well. I don't know the details of the support arrangement, but I imagine IBM will take on support specific to the code it develops.

800 pound gorilla

There is an 800 pound gorilla in the room here. What about our trusty DB2? Is it being relegated to the status of 2nd class citizen? The answer is a resounding no! DB2 is in fact one of the main reasons businesses are willing to purchase i5/OS even though its upfront costs are higher than competing solutions. This new announcement does not and cannot take away anything from DB2 and solutions using it as target database.

Ideally, mySQL brings new applications to the system. Increasing workload on the system will mean more and bigger boxes are needed and that will hopefully mean better sales news for IBM.

What's unclear

According to the announcement, mySQL will be able to access existing DB2 data and existing System i applications will be able to access new mySQL data. How exactly that's going to be accomplished is unclear to me. This is important piece that'll signify how quick and wide mySQL adoption will be in System i shops.

This announcement is fresh off the press so there is a lot of detail yet to be clarified and understood. As I learn more I'll stack it up and depending on the interest (or not) of our reader set, disseminate it in future articles.

Summary

mySQL – System i combo offers promise of bringing new application and associated workloads to our favorite platform. Now when your boss asks you why can't I find application X for the System i, chances are you'll be able to say "Wait, System i has it all".

When new application function (i.e. Query By Example) introduces a hard unpredictable workload on your system, the system administrator has to make tough decisions. For example, let's say your top notch Sales VP loves Crystal Reports and can create beautiful and useful financial reports by querying data directly via ODBC. However, he has no clue what kind of resources those queries require once they hit System i. The system administrator learns of it quickly and in excruciating detail.

Let's face it, system administrators are often low on the totem pole and don't hold much executive-level decision making power. They are first to be aware of the issue and are most likely ones that have to resolve the issue. Often, with query performance issues, performance tuning is the answer. Sometimes that is not enough. What if query Crystal Report tool generated is simply so complex, there is no way to build an index that it'll use? Or what if created report is querying entire 40 million row table to get the answer it's looking for?

Sometimes, hardware is the answer. But what if you've already maxed out your budget? Or have purchased a monster box, performed appropriate performance tuning and still have occasions when system hits 100% CPU consumption for extended time periods, slowing other, critical workload?

These are clear signals that instituting prevention policies rather than corrective actions is the right solution. You **can** control when these resource intensive queries run, at what priority, what's their time-slice once they do get processor time, what pool they run in and how much main memory they consume, do they use parallelism or not, do they run at all and much more.

You're probably thinking, what he's talking about, I don't have knobs with this level of granularity. With insure/RESOURCES you do!

This tool allows you to define policies that control all of the metrics mentioned above and implement them for specific user, group profile, *PUBLIC, certain times of day and for a specific query interface (ODBC/JDBC, RUNQRY, OPNQRYF, STRSQL and RUNSQLSTM).

What if you need to learn how users are actually using these interfaces and what kind of queries are they executing (i.e. SOX compliance)? Based on your policy configuration, insure/RESOURCES can audit every request a user makes to audit files that can be queried using built-in Crystal reports or your own user-defined reports.

Let's look at another common scenario. You've spent tens of thousands of dollars on IBM's SMP feature and only then realized that while it improves performance of certain queries tremendously, it can be a double-edged sword. While that query is running in parallel on all of your processors, it in effect monopolizes all of your CPU! You can't have that so you turn parallelism off and SMP becomes yet another shelfware.

The good news is that you can use insure/RESOURCES and selectively turn on parallelism (i.e. just for your important, long running batch jobs), thus getting return on your now shelved investment.

Get an email notification when a query that's estimated to run 5 minutes is attempted on your system. And a lot more…

Take control of your queries now. Insure/RESOURCES is the tool for that. It's a no brainer really.

Elvis

Autonomic DB2 Optimization Tool For IBM System i
New tool brings best practices to all IBM System i customers

ROCHESTER, MINNESOTA USA Mar 26, 2007 Today, Centerfield Technology, Inc. announced the development of the first autonomic tool designed to optimize DB2 for i5/OS. AutoDBA, short for the autonomic database assistant component, provides expert DB2 advice and options to automatically implement that advice. The first release will focus on fully optimizing database indices (DDS logical files and SQL indices); with the objective of making SQL and RPG programs that process large amounts of data much more efficient.

"There are literally thousands of pages of DB2 documentation. Within those pages, there are hundreds of best practices for optimizing database tables, indexes, and program access," says Mark Holm, Centerfield Technology's Chief Technology Officer. "AutoDBA is Centerfield's platform for delivering those best practices on a silver platter to shops – shops without the time or expertise to comb though a mountain of documentation. AutoDBA understands those best practices, gathers the data necessary to identify opportunities, graphically shows prioritized advice, and can optionally implement those changes with very little risk to applications."

The first autonomic tool designed to optimize DB2 for i5/OS

"AutoDBA is next step in the evolution of DB2 tools," says Elvis Budimlic, Centerfield Technology's Development Director. "The need to actively manage application databases has increased quickly in recent years due to the complexity of applications, extensive use of SQL, an explosive growth of database tables, more users, and the consolidation of servers. Unfortunately, most shops do not have extra people or time to understand the raw data available in other tools – most shops need answers and solutions, not more data."

Designed to meet the strict definition of "autonomics" as defined by IBM, AutoDBA provides granular policy controls, focused data collection, manual or automated actions (automatic

actions controlled by IT policies), full audit support, and the capability to "undo" all changes. Packaged as part of Centerfield Technology's HomeRun suite of tools, AutoDBA will be available to existing Centerfield customers by summer 2007.

About Centerfield Technology

For 10 years, Centerfield Technology has created a set of groundbreaking tools that help companies and ISVs more effectively utilize the AS/400, iSeries, and System i5 (System i). Today Centerfield helps clients manage complex databases, SQL-based applications with that tools deliver high performance database access, reduce the time needed to diagnose application issues and control unrestricted access to System i data.

Website: http://www.centerfieldtechnology.com
Centerfield Technology
Jenniefer Halverson
Director IBM System i Client Partners
email: jhalverson@centerfieldtechnology.com
phone: 507-287-8119 ext 101

Our Minnesota neighbor, TSG Server & Storage provides integrated technology solutions which include education, ERP, enterprise management and e-business.

TSG SERVER & STORAGE
Minneapolis, Minnesota
(612) 465-0801
mdubois@tsg-usa.com

Service-oriented philosophy, quality of products and outstanding commitment to the System i – AH Technology is Centerfield's exclusive distributor in Australia, New Zealand and Asia. AH Technology's mission is to provide leading-edge solutions to AS/400 users by implementing the best available operations automation and development software.

A.H. TECHNOLOGY PTY LTD
Victoria, Australia
www.ahtechnology.com.au
info@ahtech.com.au
Phone: 61-3-9571-7333

Barmor is a leading provider of comprehensive technology solutions for iSeries customers in Israel. Barmor has become a significant participant in the iSeries field in Israel, focusing on supplying clients with the latest tools and technologies available for building cutting edge solutions to fit their business needs. Centerfield is proud to introduce Barmor as our exclusive distributor in Israel.

www.barmor.co.il
info@barmor.co.il