Virtual CICS user group: Newsletter 73

Welcome to the Virtual CICS user group newsletter. The Virtual CICS user group at itech-ed.com/virtualcics is an independently-operated vendor-neutral site run by and for the CICS user community.

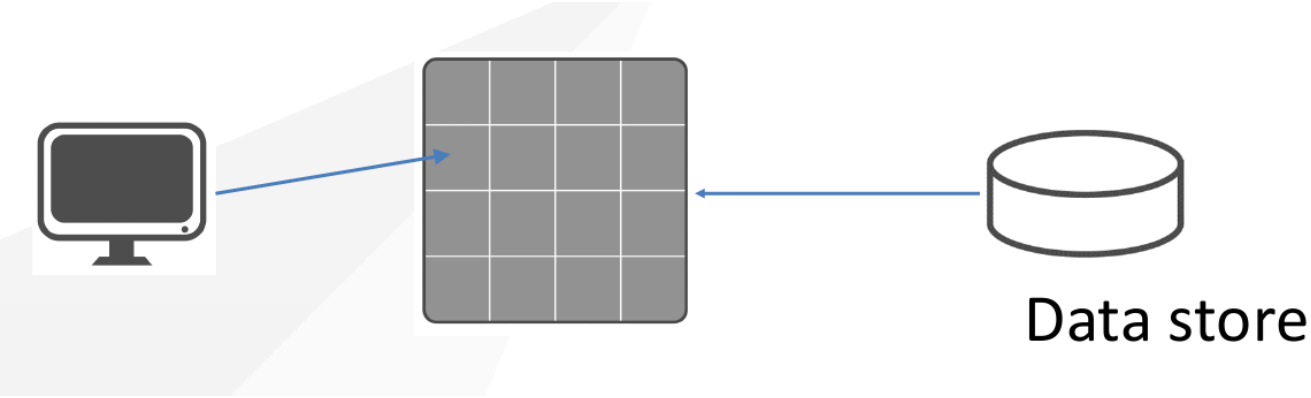

Figure 1: In-memory tables

Virtual CICS user group presentation

The latest webinar from the Virtual CICS user group was entitled, “Making the case for developers to use In-Memory tables”. It was presented in January by Larry Strickland, Chief Product Officer at DataKinetics.

Larry Strickland is the Chief Products Officer at DataKinetics. For most of the past 30 years, Larry has built his technology career by making innovative technology simple for the end user and helping them solve their most challenging issues. In fact, Larry’s long-term work with IBM and the mainframe community at large has earned him the honour of being recognized as an IBM Champion.

Larry Strickland started the presentation by saying that

Contents:

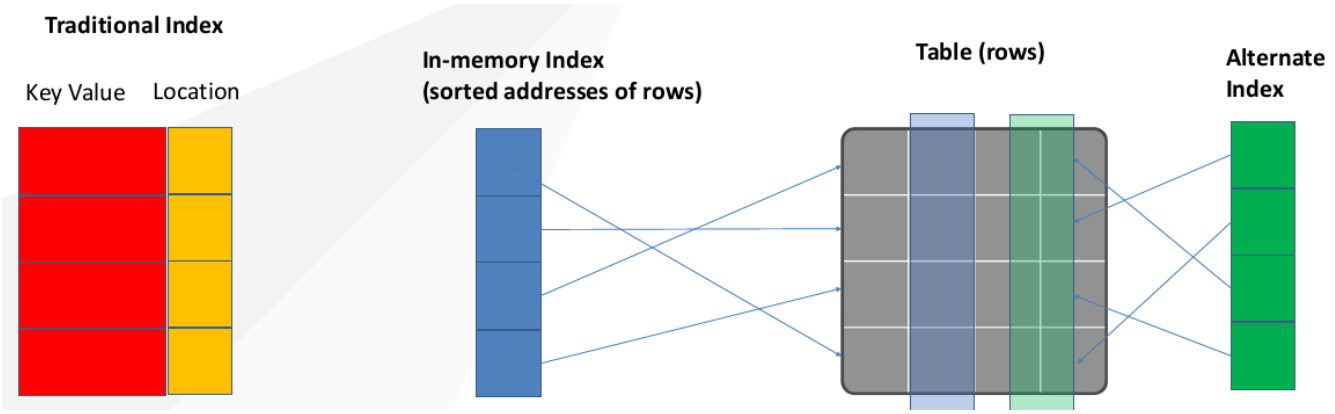

Figure 2: In-memory table indexes

although the concept of in-memory processing has been around for a long time, the falling price of RAM and growing use cases have led to a new focus on in-memory techniques and processing.

Disk access is much slower than memory access. It is orders-of-magnitude more efficient to access data from memory than it is to read it from disk. Disk I/O is an expensive operation. In addition, memory access is usually measured in microseconds, whereas disk access is measured in milliseconds (1 millisecond equals 1000 microseconds). Avoiding I/O improves performance because there is a LOT going on "behind the scenes" when you request an I/O.

Larry went on to look at different examples of in-memory use. Buffer pools basically use an agent and buffers, and may also use disk (see Figure 1).

Buffer pool pre-fetch uses a similar arrangement, but also includes a pre-fetcher. The primary goal of this technique is to reduce I/O wait times, not CPU usage. Examples would be Db2 buffer pools, package cache, instruction pipelines for CPUs, data pipelines for CPUs, and VSAM buffers.

In-memory tables are very similar except they use a data store rather than a disk. Shared in-memory tables can set read/write locks.

In-memory tables indexes are shown in Figure 2. Larry went on to explain how in-memory sorts can be used.

Db2 V12 improved its RDS sort processing using more memory. It expanded the maximum number of nodes, available in a sort tree, from 32,000 to 512,000 for non-parallel sorts or 128,000 for parallel sorts under child tasks. These enhancements might require more memory to be allocated to the thread for sort activities, but can result in a significant CPU reduction.

Although it requires the use of more memory, in-memory sorts that previously required work files for sort and merge processing are seeing a 75% reduction in CPU time. In addition, the increased sort pool size has seen a 50% reduction in elapsed time and CPU time.

In-memory hashing also shows improved performance. Larry gave some examples on in-memory tables. These included Db2, IBM Z Table Accelerator (IZTA), DKL tableBASE, COBOL (and other languages) internal tables (although limited to

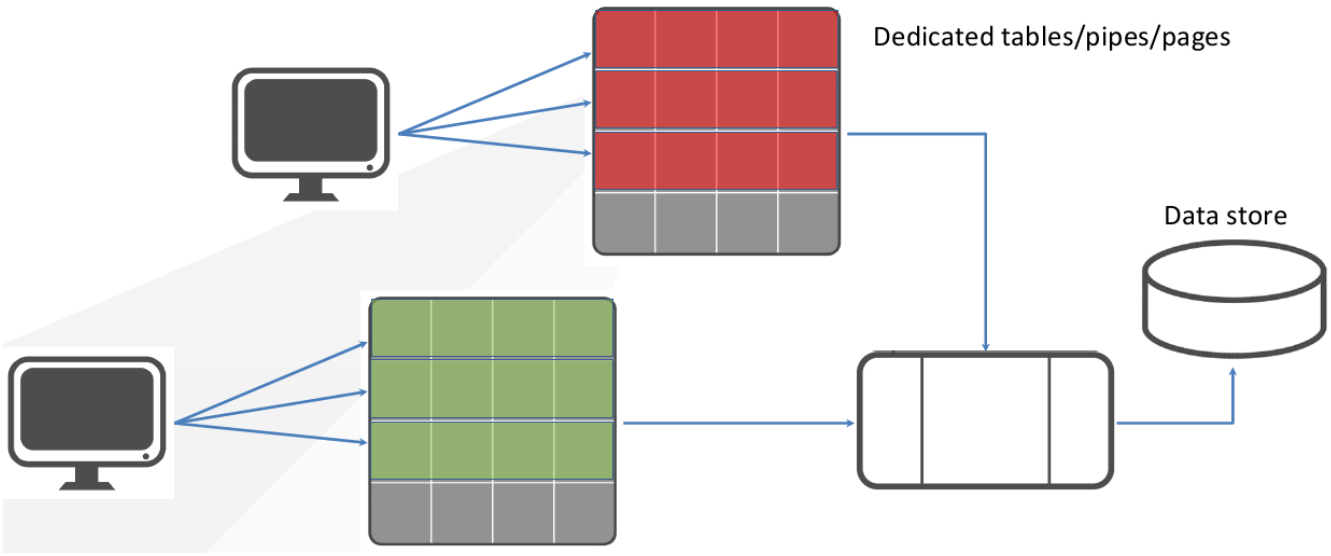

Figure 3: Fast insert

a primary index and not shareable), and some sites have their own home-grown in-memory accelerators.

Larry showed some benchmark tests illustrating savings in time and read/writes per second using in-memory processing with DKL's own product.

He went on to look at fast insert (see Figure 3), saying that they worked best where there was a high rate of concurrent INSERTs into a journal or audit table – for example with regulatory compliance or access tracking. However, there are challenges because indexing has to catch up, making immediate retrieval impossible, and keys must not conflict. Db2 v12 introduced a feature called Fast Insert 2.

Temporary in-memory tables, like other in-memory tables, require no I/O. Like fast insert they allow parallel write and no indexing. In-memory indexing is fast to create and can sort quickly as part of building the indexes.

Larry Strickland then gave some customer examples showing how they saved huge amounts of time in running applications.

He went on to suggest that it didn't matter whether a job was large or small, the advantage of in-memory tables came from using them with long-running batch jobs, processing credit cards or using frequently-opened VSAM files.

Larry moved on to explain memory usage on the mainframe. The most powerful processor chips today (including the z16) have layers of on-chip and on-board cache in the form of eDRAM and SRAM. Much of your data winds up here at some point, but it is all controlled by the system. There's not much differentiation here other than the newer chips typically have more/faster cache.

Your database is where your enterprise data lives, and where your applications go to get data. Typically, this activity requires lots of I/O disk access – this is the baseline for how fast

you access data. But since memory is about 1000x faster than disk access, you should try to use that whenever possible.

DBMS buffers use main memory to cut out I/O for recent disk access. They make a big difference in reducing data access time and processing time (which can translate to lower operations cost as well). There are even third-party buffer tools and database cache solutions that help improve buffer efficiency even more. Buffered DASD accesses data up to 10 times faster than non-buffered data.

Mainframe high-performance in-memory technology shortens the code path for the data you access most often. It augments your buffered database, using main memory, and accesses data faster than buffer performance. It requires no changes to application logic or your database. If in-memory tables are small enough, and accessed often enough, they can make it into the L3-L4 cache for ultra- fast processing.

Larry said that people often ask him about the IBM DB2 Analytics Accelerator (IDAA), which is fantastic at reducing long-running queries (by using parallelism). With this, the queries are not run often. In contrast, in-memory tables are best for reducing very short running queries, but need lots of queries before the difference is noticeable.

Larry summarized by saying that in-memory table techniques can improve the performance of specific workloads, such as:

- Buffers (or cache)
- In-memory tables
- In-memory indexes (in-memory sorts, address only changes, hash, etc)
- Fast inserts
- Temporary tables (leveraging multiple aspects)
- Small tables
- Shared tables.

A copy of Larry Strickland's presentation is available for download from the Virtual CICS user group website at https://itech-ed.com/virtualcics/presentations/CICSInMemoryMar23.pdf

You can see and hear the whole user group meeting at: https://youtu.be/frQq7fInKNs

Meeting dates

The following meeting dates have been arranged for the Virtual CICS user group:

- On 9 May we have Colin Pearce who will be discussing "How to define CICS Maintained Data Tables to the Coupling Facility Resource Manage".
- The following meeting is on 11 July, when Ezriel Gross, Principal Solutions Advisor for Rocket Software will be discussing, "Problem Analysis and Performance Tuning for CICS".

We are using Zoom for the user group meetings.

CICS articles and blogs

Lifting the Lid on CICS Temporary Storage by Andy Wright and Darren Beard in Enterprise Tech Journal issue 1 (March 2023). You can find the article at: https://mydigitalpublication.com/publication/?m=23234&i=786278&p=18&ver=html5

Detect and remediate wait problems using IBM Z OMEGAMON for CICS by Ezriel Gross in the IBM Z and LinuxONE Community

(21 February 2023). You can find the article at: https://community.ibm.com/community/user/ibmz-and-linuxone/blogs/ezriel-gross/2023/02/21/detect-and-remediate-wait-problems-in-cics

Broadcom Community Area

Don't forget the Virtual CICS User Group page in the Broadcom Commmunity area. It's a place to share ideas, collaborate, ask questions, and connect with Broadcom Mainframe Experts as well.

To register for the platform, visit https://community.broadcom.com/ and click "Register" in the upper right corner. Once registered, navigate to the Virtual CICS User Group page and click "Join".

I'll be sharing information about upcoming meetings on the page's calendar along with other information related to this user group.

About the Virtual CICS user group

The Virtual CICS user group was established as a way for individuals using IBM's CICS TS systems to exchange information, learn new techniques, and advance their skills with the product.

The Web site at itech-ed.com/virtualcics/ provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM CICS practitioners. Anyone with an interest in CICS is welcome to join the Virtual CICS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual CICS user group is free to its members.