

QuickSelect for Db2 Addresses Age-Old Db2 Performance Issue

The Opportunity

Many Db2 z/OS workloads consume a great deal of CPU resource due to certain static SQL statements retrieving the same unchanging data elements over and over. These unchanging data elements are typically found in lookup tables, references tables and the like, and may include elements such as Price, Zip Code, Fee, User Authorization, etc. When these seemingly low-cost SQLs are repeatedly executed, their aggregate impact on workload performance and CPU consumption can be very large. Because these static SQL are typically very simple, it is assumed there is no way to make them run more efficiently.

QuickSelect for Db2 is a novel approach by Log-On Software addresses this inefficiency: an external-to-Db2, detachable solution for caching static data; one that consumes less CPU and serves the cached data faster than Db2; one that is completely transparent to applications and Db2; one that guarantees 100% data integrity; one that can be easily switched on and off.

QuickSelect is employed by some of the world's largest financial institutions and telecommunications companies to dramatically reduce CPU consumption and improve the performance of Db2 applications and batch jobs.

Plug-and-Play Approach

QuickSelect is a plug-and-play solution that installs itself dynamically in a Batch or Online environment (CICS and IMS/TM). It identifies and intercepts repetitive Db2 SQL statements, caches the result sets generated by Db2 for these statements, and then for each follow-on request, serves these result sets to the requesting SQL just as Db2 would, only much faster and at a much lower CPU cost.

Absolutely no change to applications is required: no re-links, no re-binds and no JCL changes. QuickSelect is completely transparent to your applications. The only change to your application environment will be improved performance and reduced CPU consumption.

QuickSelect is completely non-invasive. Caching can be turned off with a keystroke while application programs (and all other systems) are running. The only impact, if QuickSelect is turned off, is the immediate rerouting of repetitive SQL to retrieve results sets from Db2 rather than QuickSelect cache. It is equally simple to turn QuickSelect back on to enable substantial performance gains and CPU savings.

The QuickSelect server process, implemented as a started task, caches data in self-managed memory above the bar (64 bit addressing mode) and has no impact on other commonly used storage areas.

Data Integrity and Update Sensitivity

QuickSelect's Cached Db2 SQL result sets are, of course, derived from data retrieved from one or more Db2

QuickSelect for Db2

tables. If there is a change (INSERT, DELETE, UPDATE or TRUNCATE) to data in a Db2 table underlying cached result sets, Db2 immediately notifies QuickSelect of that change via message from a VALIDPROC routine defined on the Db2 tablet. QuickSelect then immediately invalidates the cached result sets derived from the changed table(s), and repetitive SQL are rerouted to retrieve result sets from Db2 rather than cache until the cache for those results sets is rebuilt. This update sensitivity is a crucial aspect of the QuickSelect architecture.

Running Db2 utilities such as LOAD/REPLACE, REORG/DISCARD and RECOVER on a table will also cause the associated result sets in cache to be invalidated.

Finally, when QuickSelect is used in a multiple LPAR data sharing environment, update sensitivity is implemented across all LPARS using XCF functions in much the same way Db2 employs these functions.

Key Features

CPU Savings – 20% and more of the SQL a typical Db2 workload is typically diverted from Db2 to QuickSelect cache. Db2 itself no longer has to service these SQL directly. Savings of 10% or more of total CPU consumption are typically seen.

Faster Processing – Significant improvements in online application response times, and dramatic reductions in batch job elapsed time.

Application Transparency – No change whatsoever to your applications in order to achieve QuickSelect savings and benefits.

Survey Mode – QuickSelect runs a non-intrusive SURVEY in a production environment to capture statistics pointing to potential savings before caching is ever turned on!

Staggered Deployment – QuickSelect can be limited in its caching to specific programs, jobs, CICS transactions or IMS transactions (PSBs). In addition, the QuickSelect administrator has full control over which result sets (by table) may be cached by QuickSelect, thus enabling a gradual deployment.

Monitoring and Statistical Information – Extensive information about programs/tables and their cache hits is provided.

Version Compatibility

Supports Db2 V9, V10, V11, V12 all modes

Db2 data sharing is fully supported

z/OS 1.6 and up

Supports Static SQL

Saves CPU in batch jobs, CICS, IMS/TM, IMS/BMP

QuickSelect is delivered using SMP/e

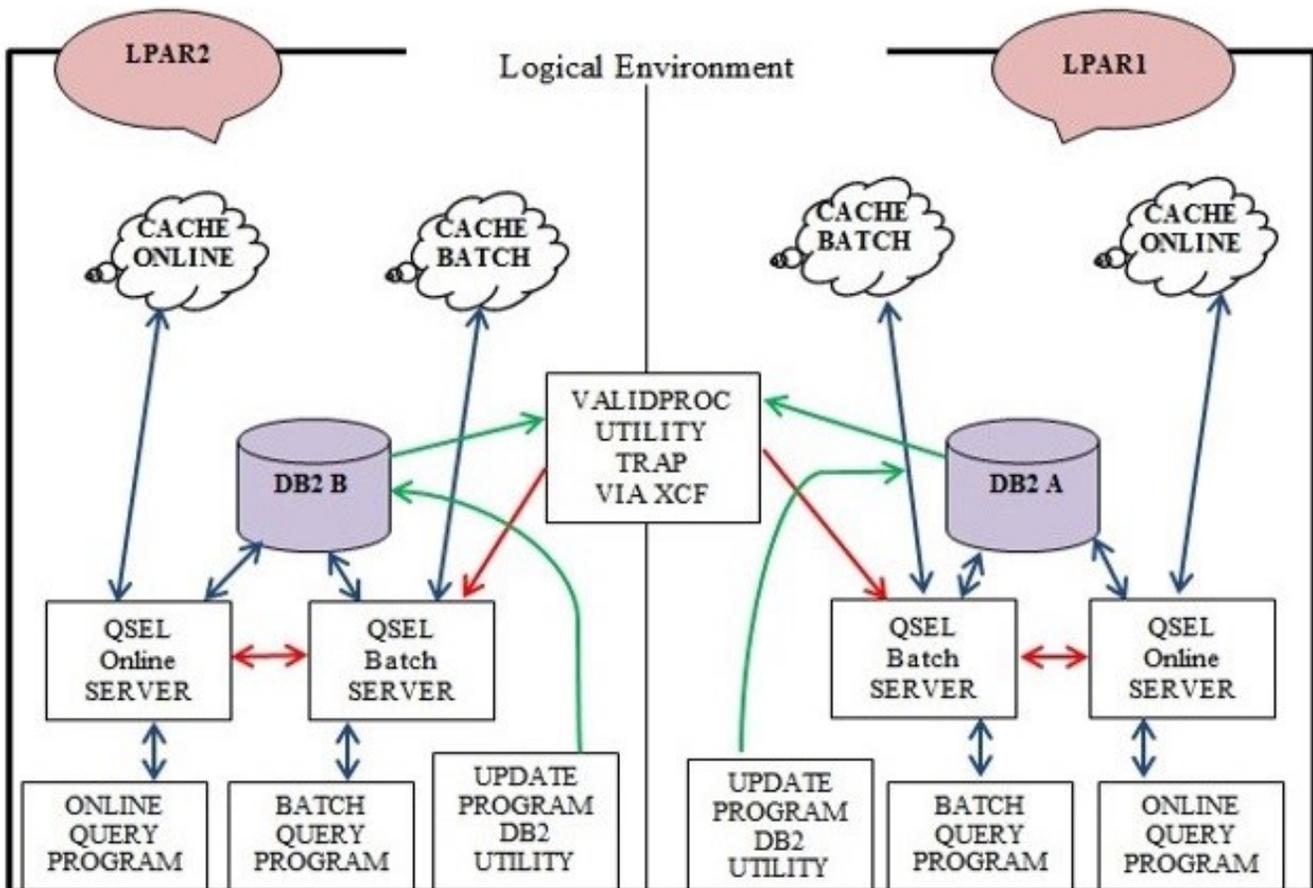
QuickSelect for Db2

Sample QuickSelect Setup

Blue Lines: QuickSelect server intercepts SQLs and loads results to cache OR retrieves results from cache and provides to application.

Green Lines: Updates from application programs and Db2 utilities.

Red Lines: QuickSelect informs all servers in an XCF group about changes to a table in order to invalidate and rebuild the cached result sets derived from that table.



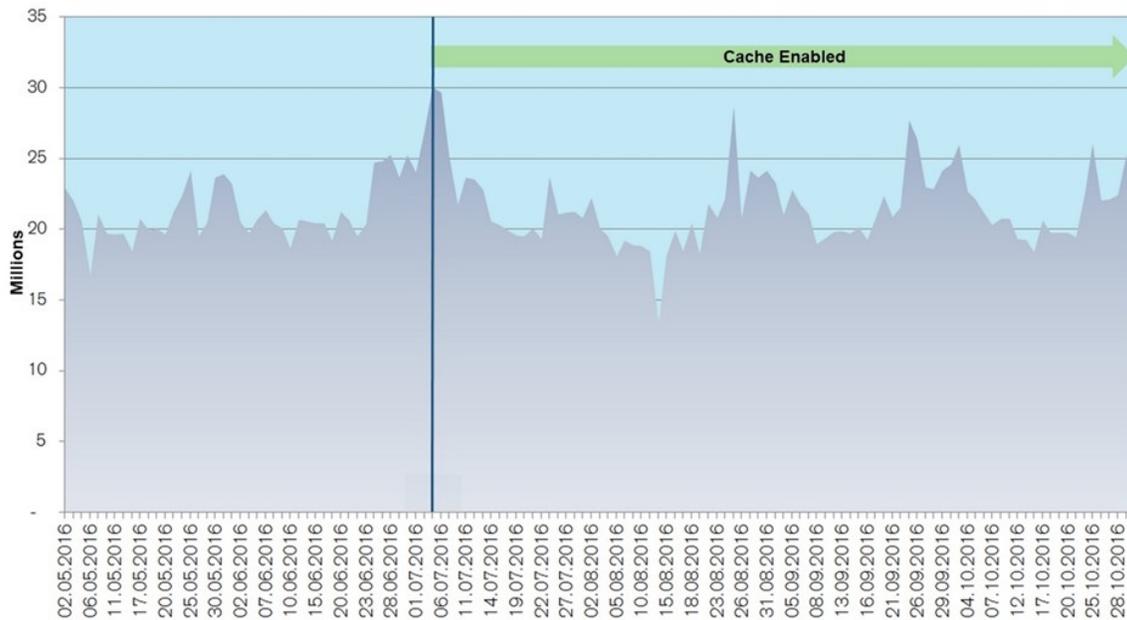
To schedule a Zoom session to discuss how QuickSelect can benefit your datacenter

Please email Christine Cargnoni at CCC@FischerInternational.com

QuickSelect for Db2

Case Study

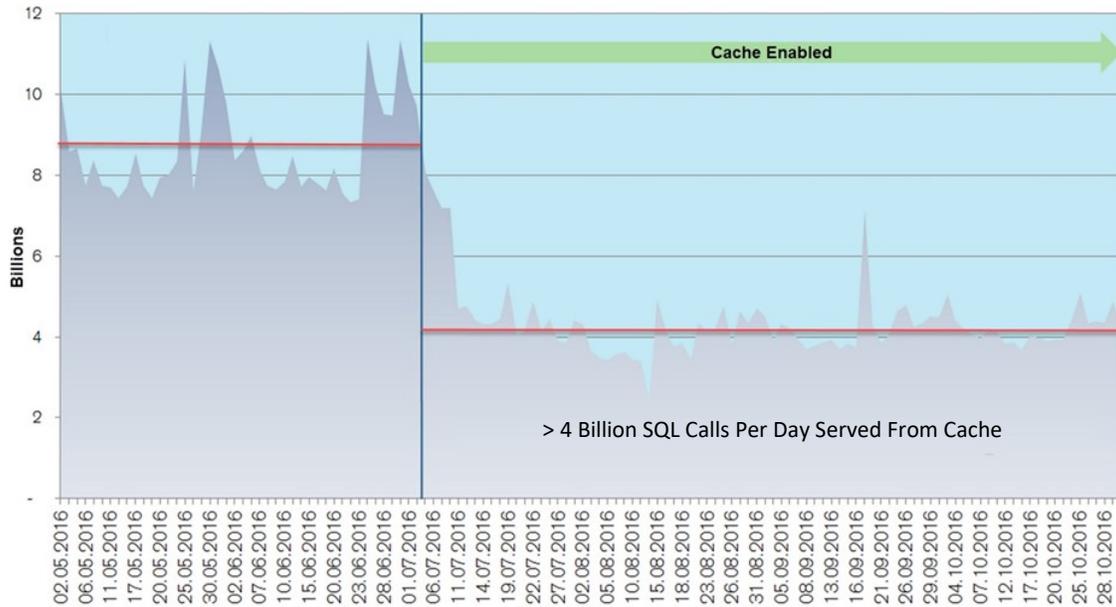
The following workload characteristic graphs reflect the positive impact of QuickSelect in the production environment of a large bank. They present a simple comparison between the performance characteristics of a Db2 workload with QuickSelect turned on and with QuickSelect turned off (remember that QuickSelect is easily turned on and off).



The graph above reflects the total workload over a period as measured by number of transactions. The blue vertical line marks the point at which QuickSelect Caching was activated. This graph shows us that the workload transaction volume before and after QuickSelect activation is the same thus assuring us that we are comparing apples to apples in our QuickSelect 'on' versus QuickSelect 'off' comparisons below.

The graph below plots the number of Db2 reads (i.e. the number of Db2 SELECT and FETCH statements) during the same time period. This graph reveals a dramatic reduction in the number of reads experienced by Db2 starting at the point of QuickSelect activation. QuickSelect identifies repetitive queries and caches their result sets. When it sees subsequent instances of the same queries, it serves the result sets directly from cache. These repetitive queries never reach Db2 resulting in the reduction in Db2 SQL executed seen below. The graph tells us that the number of repetitive queries (serviced from cache) is very high – in this case more than 4 billion per day. The number in a given environment is often much more than one might assume.

QuickSelect for Db2



The last graph plots CPU type 2 consumption (in-Db2 CPU) over the same period. It demonstrates a significant reduction of about 18% in transaction time spent in Db2. The queries serviced by Quickselect are typically simple and inexpensive, but they are highly repetitive. We see that for these SQL, serving result sets from Cache rather than Db2 results in high aggregate savings.

