How QuickSelect Solves an Age-old Problem

A considerable amount of the CPU consumption in applications is regularly spent on SQLs that are retrieving repeatedly static key data like Price Plans, Zip Codes, Fees, User Authorizations, etc. Those seemingly simple SQLs that are executed again and again during peak times, use CPU and cause higher cost– and there is no way to avoid these SQLs…

If you only had an external, detachable solution that works faster than Db2 with buffer pools, is completely transparent to applications, guarantees 100 % data integrity, can be easily switched on and off, and saves you money in the process…

Now you can implement **QuickSelect for Db2** which is based on the same principle as QuickSelect for Oracle which is used worldwide in some of the largest telecommunication billing systems with big success.

A Plug-in Solution

QuickSelect is a plug-in tool that installs itself dynamically in a Batch or Online environment (CICS and IMS/TM) and intercepts repetitive SQL statements from application programs to Db2, caching the result set and returning the same answer as Db2 would, only much faster using much less CPU.

Absolutely no change in your applications is required: no re-link, no re-bind and no JCL changes. QuickSelect in completely transparent to your applications, only that it will run much faster and use less CPU.

It is possible to turn QuickSelect off with a keystroke, while application programs (and all other systems) are running, causing all SQLs to be handled by Db2. It is equally simple to turn QuickSelect on to enable substantial CPU savings.

The QuickSelect server, implemented as a started task, caches the data in self-managed memory above the bar (64 bit addressing mode) and does not have any impact on other commonly used storage areas.

Data Integrity

Each SQL result set contains data retrieved by Db2 from one or more Db2 tables. QuickSelect caches the SQL results returned from Db2, and uses it for a repeated SQL. Since QuickSelect has to make sure that each SQL result in its cache is still valid, it must be notified about any change in any related table. Changes to data by application programs are captured by a standard VALIDPROC defined on a Db2 table that is cached by QuickSelect. Db2 automatically activates the VALIDPROC routine whenever the table contents are changed by INSERT, DELETE, UPDATE or TRUNCATE statements.

IBM Db2 utilities like LOAD/REPLACE, REORG/DISCARD and RECOVER on cached tables are caught by a utility-trap which invalidates cached data from the updated tables automatically.

QuickSelect runs also in a multiple LPAR data sharing environment and supports update sensitivity by using XCF functions in a similar way as Db2 does.

Key Features

- **CPU Savings** – Savings of 20% of the number of SQLs and 10% of total CPU at peak time are measured on average.

- **Faster Processing** – Significant improvements in online application response times, and reduction in batch jobs elapsed time are typically achieved.

- **Application Transparency** - No change whatsoever is required in your applications in order to achieve QuickSelect savings and benefits.

- **Survey Mode** - QuickSelect can run in a production environment during peak time capturing executing SQLs without a measurable impact, providing the savings potential by QuickSelect and subsequently identify which tables will be good candidates for QuickSelect caching.

- **Staggered Deployment** – QuickSelect can limit its operation to specific programs, jobs, CICS transactions or IMS transactions (PSBs). In addition you have full control on which tables will be cached by QuickSelect, thus enabling a gradual deployment.

- **Monitoring and Statistical Information -** Extensive information about programs/tables and their cache hits are provided.

Version Compatibility

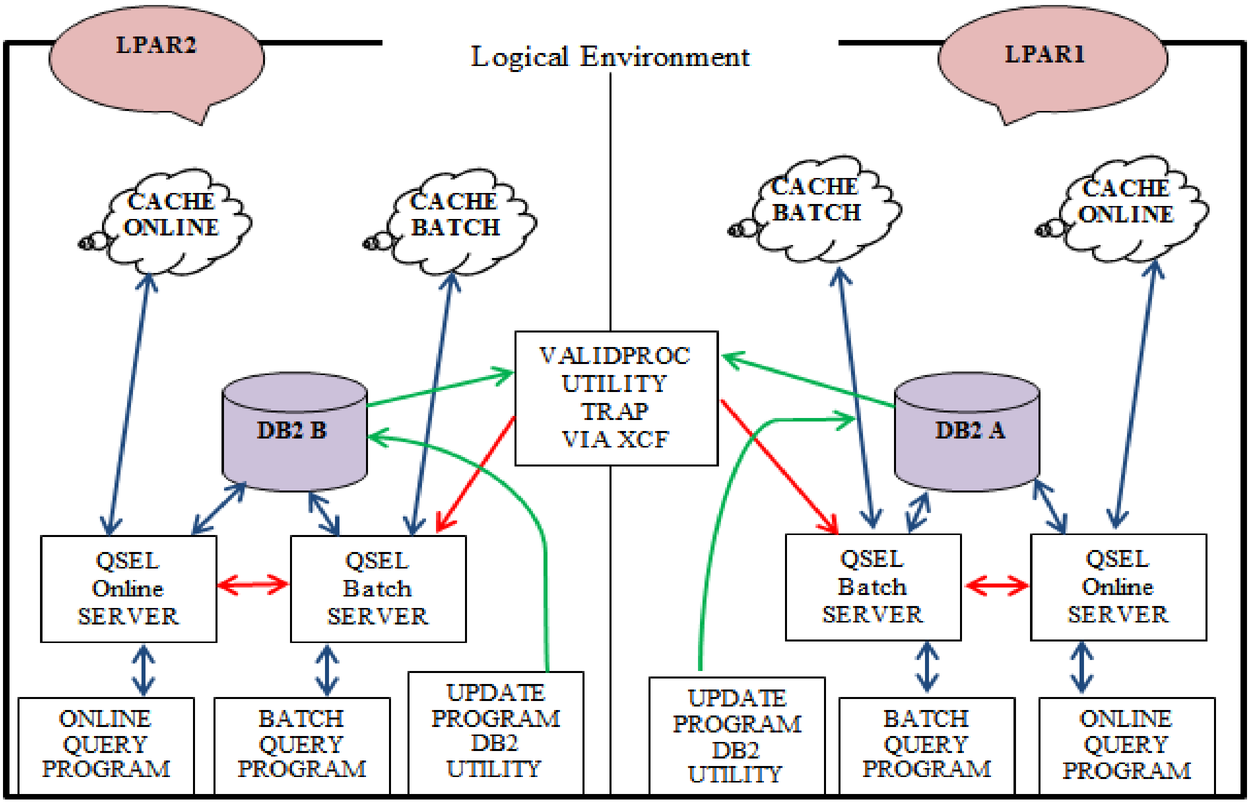
- Supports Db2 V9, V10, V11, V12 all modes
- Db2 data sharing is fully supported
- z/OS from 1.6 up (tested with z/OS 2.30)
- Supports Static SQL
- Saves CPU in batch jobs, CICS, IMS/TM, IMS/BMP
- QuickSelect is delivered using SMP/e

Sample QuickSelect Setup

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

Green Lines depict: Updates from application programs and Db2 utilities.

Red Lines depict: QuickSelect informs all servers in an XCF group about data changes in order to invalidate the cache.

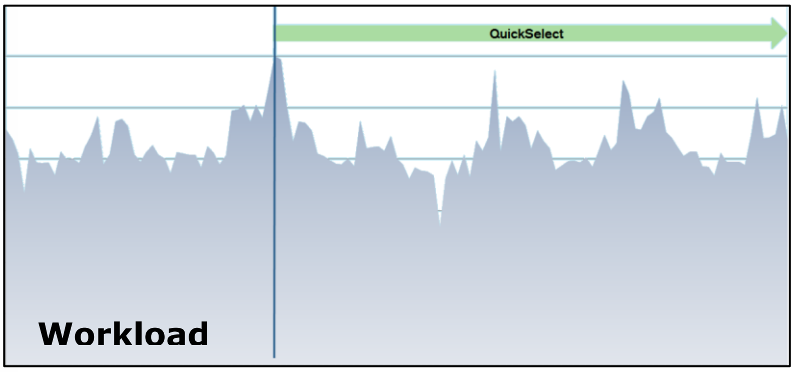


Case Study

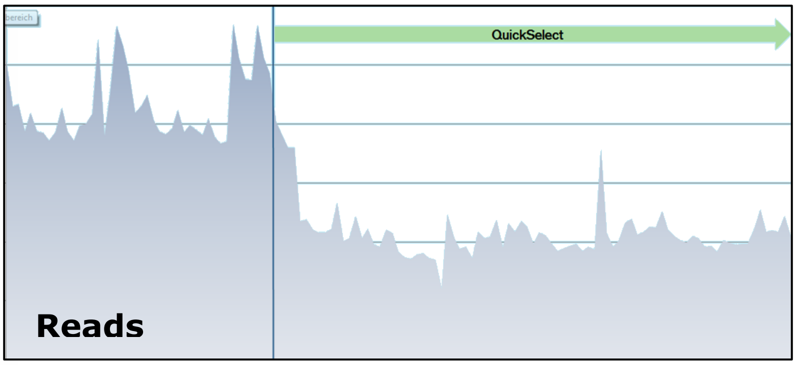
The following graphs* are results from the production environment of a large bank in Europe. The measurement took place as part of the evaluation process of QuickSelect. It is a simple comparison between working with and without QuickSelect (remember that QuickSelect is easily turned on and off).

The blue vertical line marks the time when QuickSelect was started.

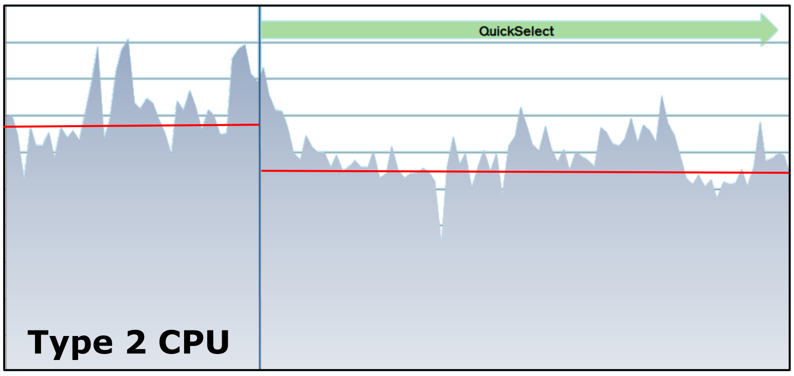
The first graph represents the total workload (measured in number of IMS transactions). It demonstrates that the workloads before and after QuickSelect activation are the same. This is to assure we compare “apples to apples” in the “before and after analysis”.



The second graph is the number of reads. That is, the number of Db2 SELECT and FETCH statements. It shows a dramatic reduction in the number of reads after QuickSelect was activated. QuickSelect identifies the repeated queries and caches their result-sets. When it recognizes subsequent activations of the same query, it provides the result from its cache. Thus, the query activation never reaches Db2, resulting in the demonstrated reduction. The graph shows that the number of repeated queries is very high. Often much more than you might assume up front. The reality shows that customers are regularly surprised from the amount of repetitive reads in their applications.



The last graph displays the measurement of the CPU type 2 consumption. It demonstrates a significant reduction in the CPU consumption, about 18% less, reducing the 4 hour Average CPU peak. Even though the saved queries are very often quick and simple, since they are repeated so many times, the reduction in the CPU consumption is dramatic.



Needless to say that this bank is now a happy customer of QuickSelect.

*) These graphs were presented by the bank DBA in an IDUG conference.