



Log-On Software QuickSelect for Db2 Introduction **A Simple to Use Plug-In Db2 SQL CPU Optimization Solution**

Value-4IT

QuickSelect for Db2 Background: Dynamic & Transparent z/OS SQL CPU Optimization

The IBM Z Mainframe server is recognised as the de facto choice for storing mission critical System of record (SOR) data in database repositories for 92 of the top 100 global banks, 23 of the 25 top global airlines; the top 10 global insurers & ~70% of all Fortune 500 companies. ~80% of mission critical data is hosted by IBM Z Mainframe servers, processing 30+ Billion transactions per day, including ~90% of all credit card transactions. This data is accessed by ~1.3 Million CICS transactions per second, compared with a Google (mostly search) processing rate of ~70,000 transactions per second.

The digital transformation has delivered an exponential growth of data, typically classified as Cloud, Mobile & Social based. This current & ever-growing data source requires intelligent analytics to deliver meaningful business decisions, requiring agile application software delivery to gain competitive edge. This digital approach can sometimes deliver a myriad of micro business application changes, personalised for each & every customer, often delivering “pop-up” applications...

Recognizing that the IBM Z Mainframe server is the de facto database server for System of Record data, primarily via the Db2 subsystem, clearly optimizing Db2 CPU usage, whether OLTP transactions, typically via CICS, or the batch window, has been & always will be, worthwhile...

From a Db2 viewpoint, pre-defined structures such as buffer pools offer some relief in storing highly referenced data in a high-speed server memory resource, but this has a finite capacity versus performance benefit, not necessarily using the fastest memory structures available nor dynamically caching the most accessed data. The business considerations of not optimizing Db2 data access are:

- **Elongated Batch Processing:** With ever increasing amounts of data to process & greater demands for 24*7*365 availability & real-time access, data access optimization is fundamental for optimized service delivery, often measured by mission critical SLA & KPI metrics. Optimized batch processing is a fundamental requirement for acceptable customer facing business service delivery.
- **Slow Transaction Response Times:** As the nature of customer requirements change, mobile device applications exponentially increasing the number of daily transactions, overall system resource capacity constraints are often stressed during peak hours. Optimized transaction response time is a fundamental requirement, being the most transparent service delivered to each & every end customer.

All too often, many IT disciplines can be classified with a generic 80/20 rule & typically data can be classified accordingly, where 80% of data is accessed 20% of the time & 20% of data is accessed 80% of the time. The challenge with such a blunt Rule of Thumb (ROT) is that it's static, but it's a good starting point. Ideally for any large data source, there would be a dynamic sampling mechanism that would identify the most active data, loading this into the highest speed memory resource to reduce I/O access times & therefore CPU usage. Dynamic management of such a data buffer would render the 80/20 rule extraneous to requirements, as each & every business has their own data access profile. However, a simple cost benefit & therefore Proof of Value (POV) analysis could ensue.

An easy but very expensive solution to remediate batch processing & transaction response issues is to provide more resources via a CPU server upgrade activity. A more sensible approach is to optimize the currently deployed resources, safeguarding that frequently accessed data is mostly if not always high speed cache resident, reducing the I/O processing overhead, reducing CPU usage, which in turn will optimize batch processing & transaction response times, while controlling associated IBM Z Mainframe server hardware & software costs.



Log-On Software QuickSelect for Db2 Introduction ***A Simple to Use Plug-In Db2 SQL CPU Optimization Solution***

Value-4IT

QuickSelect for Db2 Background: SQL Observations

The ubiquitous Db2 data access method is Structured Query Language (SQL) based, where IBM has their own implementation, SQL for Db2 for z/OS, which could be via the commonly used COBOL (EXEC SQL) programming language or a Db2 Connect API (E.g. ADO.NET, CLI, Embedded SQL, JDBC, ODBC, OLE DB, Perl, PHP, pureQuery, Python, Ruby, SQLJ). For Db2 Connect, there are 2 types of embedded SQL processing, static & dynamic SQL. Static SQL minimizes execution time by processing in advance. Though some relief is provided by Dynamic Statement Cache, dynamic SQL is processed when the SQL statement is submitted to the IBM Z Db2 server. Dynamic SQL is more flexible, but potentially slower. The decision to use static or dynamic SQL is typically made by the application programmer. There is a danger that Dynamic Statement Cache might be considered as a panacea for SQL CPU performance optimization, but as per any other performance activity, reviewing any historical changes is a good idea. The realm of possibility exists for the Db2 Subject Matter Expert (SME) to be pleasantly surprised that more often than not, there are still significant SQL CPU optimization opportunities...

From a generic Db2 viewpoint, with static SQL, you cannot change the form of SQL statements unless you make changes to the program. However, you can increase the flexibility of static statements by using host variables. Obviously, application program changes are not always desirable.

Dynamic SQL provides flexibility, if an application program needs to process many data types & structures, dictating that the program cannot define a model for each one, dynamic SQL overcomes this challenge. Dynamic SQL processing is facilitated by Query Management Facility (QMF), SQL Processing Using File Input (SPUFI) or the UNIX Systems Services (USS) Command Line Processor (CLP). Not all SQL statements are supported when using dynamic SQL. A Db2 application program that processes dynamic SQL accepts as input, or generates, an SQL statement in the form of a character string. Programming is simplified when you can structure programs not to use SELECT statements, or to use only those that return a known number of values of known types.

For Db2 data access, SQL statement processing requires an access path. The major SQL statement performance factors to consider are the amount of time that Db2 uses to determine the access path at run time & whether the access path is efficient. Db2 determines the SQL statement access path either when you bind the plan or package that contains the SQL statement or when the SQL statement executes. The repeating cost of preparing a dynamic SQL statement can make the performance worse when compared with static SQL statements. However, if you execute the same SQL statement often, using the dynamic SQL statement cache decreases the number of times dynamic statements must be prepared.

Typically, organizations have embraced static SQL over dynamic because static is more predictable, showing little or no change, while dynamic implies ever changing & unpredictable. Db2 performance optimization functions have been incorporated into base Db2 (E.g. Buffer Pools) & software products (E.g. IBM Db2 AI for z/OS, IBM Db2 for z/OS Optimizer, IBM Db2 Analytics Accelerator, IBM Z Table Accelerator, IZTA), with varying levels of benefit & cost. Ultimately IBM Z Mainframe customers need simple cost-efficient off-the-shelf solutions of a plug & play variety & without doubt, optimizing static SQL data processing is a pragmatic option for reducing Db2 subsystem CPU usage.

As we have identified, Db2 can consume significant amounts of z/OS CPU accessing & retrieving the same static frequently used data elements repetitively. Upon analysis, these static frequently used data elements are typically identified originating from a small percentage of Db2 tablespaces. Typically, at first glance these simple SQL programs are considered as low risk, but are repeatedly processed, often in peak processing times, consuming excessive CPU & increasing processing cost accordingly, typically z/OS Monthly Licence Charges (MLC) related. Db2 optimization tools for access path or buffer pool management provide some benefit, but this is not always significant & may require application changes. Patently there is a clear & present requirement for a simple plug & play solution, transparent to Db2 processing, maintaining an optimized high-performance in-memory cache of frequently used Db2 data, safeguarding data integrity in environments various, including SYSPLEX, Data Sharing, et al...



Log-On Software QuickSelect for Db2 Introduction **A Simple to Use Plug-In Db2 SQL CPU Optimization Solution**

Value-4IT

QuickSelect for Db2 Introduction: A Minimal Overhead Plug-In & Easy-To-Use Solution

QuickSelect is a plug-in solution dynamically activated in a batch or OLTP environment (I.E. CICS, IMS/TM) intercepting repetitive SQL statements from Db2 application programs, storing the most active result set, not necessarily the entire tablespace, in a high-performance in-memory cache, returning to applications the same result set as per Db2, but much faster & using less CPU accordingly. QuickSelect is completely transparent to z/OS applications, eliminating any requirement to change/recompile/relink application source or rebind packages. QuickSelect processing can be switched on or off using a single keystroke, either defaulting to standard Db2 SQL processing or to benefit from the QuickSelect high-speed cache for optimized CPU resource usage.

The 64-bit QuickSelect server, implemented as a started task, intelligently caching data in self-managed memory above the bar, supporting up to 16 EB of memory, eliminating concerns of using any other commonly used storage areas (E.g. ECSA). The intelligent caching mechanism safeguards that only highly active data is retained, optimizing the associated cache memory size required.

QuickSelect caches frequently requested Db2 SQL result sets, returning these results to the application from QuickSelect cache, when a repetition of the same SQL is encountered. For data integrity purposes, QuickSelect immediately invalidates result sets upon detection of changes to underlying tables, implicitly validating each cache resident SQL result set. Changes to Db2 data by application programs are captured by a standard Db2 VALIDPROC process, attached to the typically small subset of frequently accessed tables of interest to QuickSelect. Db2 automatically activates the VALIDPROC routine whenever the table contents are changed by INSERT, DELETE, UPDATE or TRUNCATE statements, invalidating cached data from the updated tables automatically. For standard Db2 utilities such as LOAD/REPLACE, REORG/DISCARD & RECOVER, table-level changes are identified by a QuickSelect utility-trap, invalidating cached data from the updated tables automatically. QuickSelect also supports SYSPLEX & Data Sharing environments, supporting update activity via the same XCF functions & processes used by Db2.

QuickSelect for Db2 Compatibility: Db2, z/OS & IBM Z Server

QuickSelect is a 64-bit software solution, seamlessly integrating with all z/Architecture environments:

- **Db2 Compatibility:** Always supports the latest supported Db2 versions in all modes, with back-level support back to Db2 for z/OS Version 9.
- **z/OS Compatibility:** Always supports the latest supported z/OS versions, with back-level support back to z/OS Version 1.6.
- **IBM Z Server Compatibility:** Always supports the latest supported IBM Z servers, with back-level support for all z/Architecture servers.
- **Db2 Operating Modes:** z/OS Parallel Sysplex, XCF & Data Sharing.
- **Db2 for z/OS SQL:** Supports static SQL.
- **Workload Support:** OLTP via CICS & IMS/TM, batch SQL processing (E.g. COBOL, QMF, SPUFI, IMS BMP).
- **Installation:** Full SMP/E installation methodology.



Log-On Software QuickSelect for Db2 Introduction **A Simple to Use Plug-In Db2 SQL CPU Optimization Solution**

Value-4IT

QuickSelect for Db2: Benefits

QuickSelect for Db2 delivers the following benefits:

- **CPU Savings:** Meaningful reduction (E.g. 20%) in the Db2 SQL direct processing; 10%+ peak time CPU reduction is not uncommon.
- **Faster Processing:** Optimized CPU usage delivers shorter batch processing & OLTP transaction response times, for related SLA & KPI objective compliance.
- **Transparent Implementation:** No application changes required, source code, load module or Db2 package.
- **Survey Mode:** Unobtrusive & minimal Db2 workload overhead data sampling to identify potential CPU savings from repetitive SQL & tables of interest, before implementation.
- **Staggered Deployment:** Granular criteria (E.g. Job, Program, Table, Transaction, Etc.) implementation ability.
- **Reporting & Analytics:** Extensive information detailing cache usage for Db2 programs & tables.

QuickSelect for Db2: Management Summary

Since 1993 Db2 has evolved dramatically, in line with the evolution of the IBM Z Mainframe server. When considering today's requirement for a digital world, processing ever increasing amounts of mission critical data, a base requirement to optimize CPU processing for Db2 SQL data access is mandatory. In a hybrid support environment where today's IBM Z Mainframe support resource requires an even blend of technical & business skills, plug & play, easy-to-use & results driven solutions are required to optimize CPU usage, transparent to the subsystem & related application programs. QuickSelect is such a solution, fully exploiting 64-bit z/Architecture for ultimate scalability, identifying & resolving a common CPU consuming data access problem, for a mission critical resource, namely the Db2 subsystem, maintaining mission-critical System of Record data.

z/OS CPU optimization is a mandatory requirement for every organization, to reduce associated software & hardware costs & in theory, as a mandatory pre requisite for deploying the Tailored Fit Pricing for IBM Z pricing mechanism. Tailored Fit Pricing uses the previous 12 Months SCRT submissions to establish a baseline for MSU charging over a contracted period, typically 3 years. If there are any unused MSU resources, these are carried forward to the next year, but if those MSU resources remain unused at the end of the contracted period, they are lost, meaning the organization has paid too much. If the MSU resource exceeds the agreed Tailored Fit Pricing baseline, excess MSU resources are charged at a discounted rate. Clearly achieving an optimal MSU baseline before embarking on a Tailored Fit Pricing contract is arguably mandatory & it therefore follows that optimizing CPU forever more, safeguards optimal z/OS software charging during the Tailored Fit Pricing contract. QuickSelect for Db2 is a seamless CPU optimization product that will perpetually deliver benefit, assisting organizations to minimize their z/OS MLC costs, whether they continue to proactively manage the R4HA, submitting monthly SCRT reports or they embark on a Tailored Fit Pricing contract...

For more information please visit our [QuickSelect Portal](#), Email our [Sales Team](#) or call us on +44 (0) 845 0579386.

Value-4IT Limited
7 Wright Road, Long Buckby
Northampton, NN6 7GG
United Kingdom
Tel: +44 (0) 845 0579386
sales@value-4it.com
www.value-4it.com



Log-On Software Ltd
3 Hachilazon Street
Ramat-Gan 52522
Israel
Tel: +972-3-576-3100
info@log-on.com
www.log-on.com