Sistemas aKúbica

POWERED BY Progress

THE GREAT PERFORMANCE OF PROGRESS OPENEDGE 12 WITH SSJ (SERVER-SIDE JOIN)

Sistemas aKúbica

POWERED BY Progress

THE GREAT PERFORMANCE OF OPENEDGE 12 WITH SERVER-SIDE JOIN

Are you aware of the new Server-Side Join (SSJ) introduced in OpenEdge version 12.0?

The SSJ functionality was introduced in OpenEdge 12.0 with the FOR statement, now with 12.1 you can use it with dynamic FOWARD-ONLY, NO-LOCK queries. In this tutorial we will show you how to implement this functionality.

The multi-threaded Database Server of OpenEdge 12 processes remote client requests at the same time, improving DB performance significantly, without requiring changes to application coding. For better performing queries OpenEdge 12 performs Server-Side Joins with FOR-EACH queries that join multiple tables. These queries are now resolved server-side rather than client-side, and require no code changes.

SSJ supports up to 10 tables joined in the same logical DB. When a query cannot be resolved on the server side it continues to work as usual.

Applications using an n-tier architecture connect to the DB via network parameters. In this scenario the client can be a GUI, TTY, or a PASOE process agent.

Finally, SSJ processing improves performance by resolving queries on the server and reducing data sent over the network, and is default for joins that meet all of the following conditions:

1. Joins in a client / server environment.
2. Bind using NO-LOCK with the FOR statement, or the FORWARD-ONLY dynamic query.
3. Joins up to 10 tables in the same logical DB.

It is implemented as follows:

PROSERVE PARAMETERS

The SSJ functionality is controlled by the DB parameter -ssj. By default the database uses -ssj 1. The use of SSJ can be disabled by specifying -ssj 0 as the startup parameter when starting the broker with PROSERVE. MTS (multi-threaded server) is also enabled from the start.

Parameters	Server-Side Join
-threadedServer 1 -ssj 1	Enabled (default) Multi-threaded server (MTS) with server-side join (SSJ)
-threadedServer 1 -ssj 0	Disabled
-threadedServer 0 -ssj 1	-ssj 1 is ignored because the multi-threaded server is not enabled.
-threadedServer 0 -ssj 0	Disabled

SSJ PROCESSING WITH A FOR STATEMENT

This program shows a join using the FOR statement. The connection to the DB is made via network parameters.

These are the steps to run an example program using SSJ on a system with OpenEdge 12.1 installed:

1. prodb sports2020 sports2020
2. proserve sports2020 -S 20000
3. mpro sports2020 -S 20000 -p SSJ1.p

The connection to the database is done using the network parameter "-S 20000". The client / server connection is established. The SSJ1 .p program runs and displays the elapsed time for the query.

Notes:

- The performance depends on the data and the reduction of the data sent over the network.
- There are no changes to the index selection rules when you run the SSJ functionality.
- If needed, you can specify USE-INDEX to ensure that a certain index is used for optimal query resolution.

PERFORMANCE TEST DISABLING SSJ

To compare the execution speed of the program by disabling SSJ you can do the following:

1. prodb sports2020 sports2020 -ssj 0
2. proserve sports2020 -S 20000
3. mpro sports2020 -S 20000 -p SSJ1.p

The difference with the previous steps is that the option “-ssj 0” disables SSJ.

When you run the performance test, you will need to close and restart the DB broker so that the query enhancements are not associated with the caching of blocks or records.

Sistemas aKúbica

POWERED BY Progress

SAMPLE CODE USING FORWARD-ONLY DYNAMIC QUERIES

SSJ2.p

```
define variable qh as widget-handle no-undo.
define variable lSuccess as logical no-undo.

etime(yes).
output to report.txt.

create query qh.

qh:forward-only = true.
qh:set-buffers(buffer customer:handle,
               buffer order:handle,
               buffer orderline:handle).

qh:query-prepare(
    "for each customer no-lock," +
    "    each order no-lock " +
    "        where order.custnum = customer.custnum" +
    "          and promisedate = 05/28/2018," +
    "    each orderline no-lock " +
    "        where orderline.ordernum = order.ordernum"
).
qh:query-open.

repeat while not qh:query-off-end:
    lSuccess = qh:get-next().
    if lSuccess then
    do:
        put customer.custnum format ">>>>>9"
            order.ordernum " "
            orderline.linenum skip.
    end.
end.

qh:query-close().
delete object qh.

output close.
display proversion etime.
pause.
quit.

```

1. prodb sports2020 sports2020
2. proserve sports2020 -S 20000
3. mpro sports2020 -S 20000 -p SSJ2.p

This example program uses FORWARD-ONLY Dynamic query with NO-LOCK.

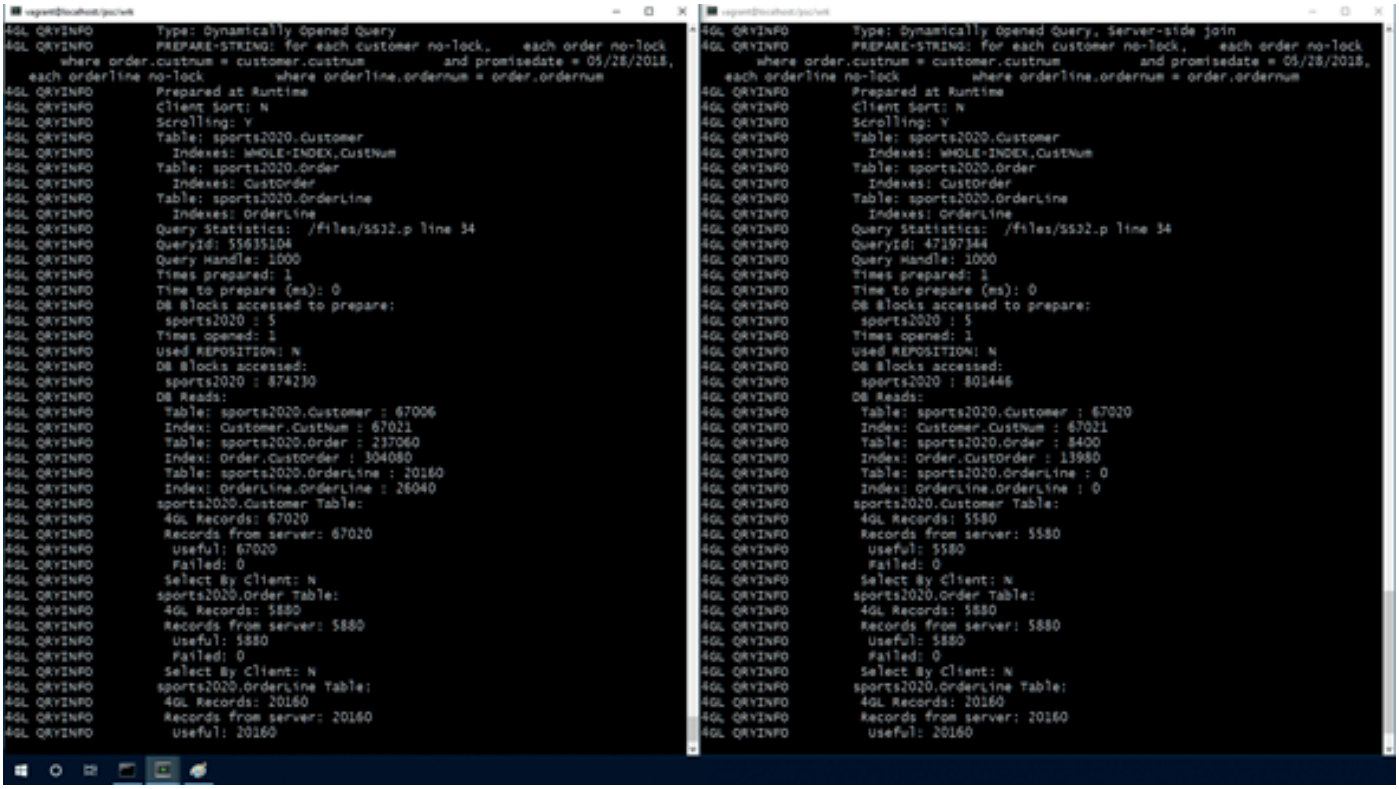
QRYINFO LOGGING

You can analyze the execution of a query using QryInfo logging option without SSJ, as follows:

1. proserve sports2020 -S 20000 -ssj 0
2. mpro sports2020 -S 20000 -p SSJ2.p \
   -clientlog client.log -logentrytypes QryInfo -logginglevel 3

To run the example using QryInfo logging with SSJ:

1. proserve sports2020 -S 20000
2. mpro sports2020 -S 20000 -p SSJ2.p \
   -clientlog client.log -logentrytypes QryInfo -logginglevel 3



In this screenshot you can see the log using QryInfo, the left side shows the result with the SSJ option disabled, and the right side shows SSJ enabled (default).

Sistemas aKúbica

POWERED BY Progress

CONCLUSION

The SSJ processing functionality in OpenEdge 12.x delivers a huge improvement out-of-the-box performance.

To remember:

- SSJ is enabled by default.
- SSJ is available with the FOR statement and FORWARD-ONLY dynamic queries, NO-LOCK.
- Actual performance gains depend on the data being queried and the reduction of the records sent over the network.
- The QryInfo logging option can be used to analyze the query execution.

¿ESTÁS LISTO PARA DAR EL SIGUIENTE PASO?

Consulta con nosotros todas las posibilidades de hacer crecer la innovación en tu empresa

Alejandro López
81 1077 0559
alopez@akubica.com

Arantza Guajardo
81 1965 5289
aguajardo@akubica.com

contacto@akubica.com

www.akubica.com

/sistemasakubica

company/sistemas-akúbica

Fuente: https://www.progress.com/blogs/openedge-12-great-performance-out-of-the-box-with-server-side-join