MarkLogic Server

Release Notes Addendum

MarkLogic 10
May, 2019

Last Revised: 10.0-7, May, 2022

1.0 Improved Streaming for XQuery FLWOR Expressions

XQuery FLWOR expressions that only use "let" will now stream the results. Prior to MarkLogic 10.0-7, they would have been buffered in memory. This allows large result sets to be more easily streamed from XQuery modules.

Due to this change, code that relied on the previous behavior of buffered results from FLWOR expression with only a "let", may perform worse if the results are iterated over multiple times. This is due to the fact that once a streaming result has been exhausted, the query has to be rerun to iterate over it again.

Even prior to this change, it is best practice to treat all query calls as lazily-evaluated expressions, and only iterate over them once. If the results need to be iterated multiple times, wrap the search expression in `xdmp:eager()` or iterate over the results once and assign that to a new variable.

For example, in MarkLogic 10.0-7 and prior versions, the following expression would be lazily-evaluated and run the search multiple times, if the `$results` variable is iterated over multiple times.

```
let $_ := xdmp:log("running search")

let $results := cts:search(fn:collection(), cts:word-
query("MarkLogic"))
```

This behavior has not changed in MarkLogic 10.0-7. However, prior to MarkLogic 10.0-7, the following expression would short-circuit the lazy evaluation and buffer all of the results in memory.

```
let $results :=
    let $_ := xdmp:log("running search")
    return cts:search(fn:collection(), cts:word-query("MarkLogic"))
```

In MarkLogic 10.0-7, this behavior is now consistent with the other form of the expression above and returns an iterator. The search will be run multiple times if the `$results` variable is iterated over multiple times.

To achieve the same buffering behavior in MarkLogic 10.0-7, wrap the `cts:search()` call in `xdmp:eager` (https://docs.marklogic.com/xdmp:eager) as follows:

```
let $results :=
    let $_ := xdmp:log("running search")
    return xdmp:eager(cts:search(fn:collection(), cts:word-
query("MarkLogic")))
```

To help understand if a variable will stream or not, the `xdmp:streamable` (https://docs.marklogic.com/xdmp:streamable) function was also added in MarkLogic 10.0-7.

For more information about lazy evaluation in MarkLogic, see the following resources:

- https://www.marklogic.com/blog/lazy-evaluation/
- https://www.marklogic.com/blog/lazy-vs-eager-evaluation/