Single Instruction Multiple Data (SIMD) is a type of parallel processing, where a single instruction is applied on multiple data. This instruction is not an aggregation, which means as you can see in Figure on the right that n-Input on a single instruction should produce n-outputs.

High throughput is one of the biggest advantages of SIMD and is needed for in-memory data storage. To realize in-memory data storage the concept of SIMD is implemented with vectors as the format. To apply the SIMD concept to this format, different types of algorithms can be used.

Run-Length Encoding (RLE) is one of those algorithms, which counts the number of occurrences of the same value. Each Run is represented by the value and number of occurrences. Most variations of the algorithm have been proposed with 128-Bit SIMD registers, which means for an element size of 32-Bit, the algorithm can process 4 elements at a time. But the concept of the algorithms allows the usage of different sizes of SIMD registers, like 256-Bit or 512-Bit.

**Example run with 128-bit register**

**Evaluation for different register sizes**

Measured in million integers per second

Register sizes compared with 128-bit as the baseline

Close up for the non-well performing areas