Single Instruction Multiple Data with Run-Length-Encoding

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 the same value. Each Run occurrences of 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.



SIMD

Instructions

Data

Result

Example run with 128-bit register



Evaluation for different register sizes



Close up for the non-well performing areas



David Matuschek

Bachelor Student IT-Systems Engineering Hasso Plattner Institute, Potsdam, Germany E-Mail: david.matuschek@student.hpi.de

Lecture Series on Database Research

Based on the talk of Dr. Wolfgang Lehner

References:

[1] https://www.tele-task.de/lecture/video/8874/[2] https://johnysswlab.com/crash-course-introduction-to-

parallelism-simd-parallelism/

