Efficient SIMD Vectorization for Hashing in OpenCL
Tobias Behrens\textsuperscript{1}, Viktor Rosenfeld\textsuperscript{1}, Jonas Traub\textsuperscript{2}, Sebastian Breß\textsuperscript{1,2}, Volker Markl\textsuperscript{1,2}

\textsuperscript{1}firstname.lastname@dfki.de
\textsuperscript{2}firstname.lastname@tu-berlin.de

\begin{abstract}
Hashing is at the core of many efficient database operators such as hash-based joins and aggregations. Significant speedup was shown for vectorized hash table operations using processor specific low-level intrinsics. We present portable and vectorized hashing primitives using the parallel programming framework OpenCL.
\end{abstract}

\begin{vectorized}
\textbf{Vectorized Data Movement Primitives}

\begin{itemize}
\item \textbf{Selective Load} Copies data from contiguous memory into selectable SIMD lanes.
\item \textbf{Selective Store} Copies data from selectable SIMD lanes to contiguous memory.
\item \textbf{Gather} Copies data from discontiguous memory into SIMD lanes.
\item \textbf{Scatter} Copies data from SIMD lanes into discontiguous memory.
\end{itemize}

\textbf{Gather OpenCL} vs. \textbf{Gather Intel Intrinsics}

\begin{itemize}
\item \textbf{Portable and Maintainable Code}
\end{itemize}

\begin{performance}
\textbf{Performance on Xeon CPU}

\begin{itemize}
\item \textbf{Build} is overhead dominated, OpenCL-based probe outperforms scalar implementation.
\item \textbf{Intrinsic-based implementation outperforms OpenCL-based on Xeon Phi.}
\end{itemize}

\textbf{Take Home}

\begin{itemize}
\item OpenCL reduces code complexity and ensures portability of vectorized primitives.
\item OpenCL-based vectorized hashing outperforms scalar hashing on Xeon CPU.
\item Processor specific intrinsics are still faster, especially on Xeon Phi.
\end{itemize}

\textbf{Funding Acknowledgements}

This work was funded by the EU projects SAGE (671500) and E2Data (780245), DFG Stratosphere (606902), and the German Ministry for Education and Research as BBDC (01IS14013A) and Software Campus (01IS12056).

\textbf{Open Source Repository}

github.com/TU-Berlin-DIMA/OpenCL-SIMD-hashing