Motivation & Contributions
- Kokkos provide tuning parameters such as the size of thread teams and vector width.
- For many applications, choosing the right parameters can be highly dependent on the data input or application configuration.
- SpMV performance is often highly correlated with the associated matrix sparsity.
- Extend the current set of Kokkos profiling tools with an online-autotuner that can iterate over possible:
  - Thread team size
  - Vector width
- Evaluate the autotuner on the latest classes of HPC devices - NVIDIA’s Pascal GP100, and Intel’s Knights Landing (KNL).

Kokkos
- Programming model that abstract code from the finer intricacies of hardware details.
- Provides abstractions for the memory space and execution space.
- Parallel patterns:
  - parallel-for, reduce, and -scan
- Abstract machine model
  - Multiple execution and memory spaces.

Accelerators and Matrices
- Each architecture has a substantially different set of optimal parameters.
- The two KNL configurations have the same parameters less than half the time.
- The algorithm is defined in line 8 of code listing.
- The number of non-zeros increases as a hint.
- The program computes blocks of the overall size in parallel.
- The number of non-zeros on each row increases.
- The number of non-zeros increases.
- Not particularly true for KNL Alpha with HBM enabled as it may depend more heavily on other matrix features such as memory size.
- Portable across and within architectures has to be done by taking into account small hardware details and data characteristics.

SPVM Skeleton code
- The algorithm is defined within a C++ functor from line 19 to 32.
- Series of nested parallel patterns in lines 19 and 22.
- Functor is instantiated in line 5 and used as the last parameter for the parallel pattern in line 11.
- Breaking the problem size into groups of team threads and specifies the vector width.

Pascal Results
- Fixed:
  - Large irregular matrices perform poorly due to divergence.
  - Smaller irregular matrices and denser matrices perform almost optimally.
  - By around 10 non-zero elements’ divergence is not longer an issue.
- Autotuner:
  - 5X to 5X faster than fixed for large irregular matrices and only 12% to 15% slower than Oracle.
  - As the number of non-zero increase performance is only marginally better than Fixed.
  - For the denser type of matrices sub-optimal choices by the autotuner suffer substantial penalties.

Conclusion & Future work
- Extended the Kokkos performance tools with an autotuner that iterates over possible candidate parameters.
- Compared the autotuner against a Fixed approach and the Oracle on the latest two distinct accelerator architectures available to this date.
- Identified matrix characteristics that affect the performance of the autotuners.
- Plan to augment the current autotuner with capabilities to extract information about the matrix and prune the search space using more advance heuristics.

Acknowledgments