Porting SEISCOPE kernels of elastic VTI wave propagation & extraction in frequency to Intel Knights Landing

SEISCOPE(*) is a consortium managed by the 3 French public laboratories LJK, Geoazur and ISTERRE, and sponsored by 9 Oil & Gas companies working together in quantitative seismic imaging with the aim to reap the outcome of their common R&D endeavor to enhance their operations.

In the SEISCOPE framework an efficient 3D finite-difference time-domain modelling and frequency-domain inversion code of Full Waveform Inversion called GeoInv3D is developed. GeoInv3D is memory bounded and is expected to take benefit from Intel High Memory Bandwidth MCDRAM that equips the last generation Xeon Phi™ Knights Landing (KNL).

Atos Center for Excellence in Parallel Programmation (CEPP) presents results obtained on an Intel Xeon Phi™ KNL (7210).

Specifications:

Runs are performed with dual-socket Broadwell Intel® Xeon™ E5-2680 v4 (**BWD**) and 1 socket "Knights Landing" Intel® Xeon Phi™ 7210 (**KNL**), respectively with 2x14 cores at the nominal frequency 2.40 GHz and 64 cores at the nominal frequency 1.30 GHz. BDW is equipped with 128 GB DDR4 and KNL with 96 GB DDR4 & 16 GB Multi-Channel DRAM (MCDRAM). Here we use the KNL with the quadrant clustering mode and MCDRAM is used either as a last-level cache (cache mode) or as addressable memory (flat mode).

GeoInv3D is used in Single Precision. It is a hybrid code using MPI & OpenMP with 3 levels of parallelism on sources distribution (MPI), domain decomposition (MPI) and shared-memory parallelism of the expensive kernels (OpenMP): here we do not use sources distribution (one explosive source) but consider domain decomposition in our results(**). The discretization is made with a staggered grid on which Perfectly Matched Layer boundary conditions are applied.

Benchmarks:

To reduce total runtime and focus of relevant parts of the code, only modelling is considered. Thus we look at the 3 main (from a computing point of view) kernels involved in the wave propagation plus one that we add to mimic the cost of an inversion in frequency:

- **Kernel A**: COMPUTE_ELAS_NORMAL_STRESS ;
- **Kernel B**: COMPUTE_ELAS_SHEAR_STRESS ;
- **Kernel C**: COMPUTE_ELAS_VELOCITY ;
- **Kernel D**: EXTRACT_FREQ_WAVEFIELD_FDM.

Five benchmarks were considered in this work: first a little benchmark **LBen**, then 3 medium benchmarks **M1Ben**, **M2Ben** & **M3Ben** with growing memory footprint, and finally a big benchmark **BBen**. For each benchmark the simulation lasts 4 seconds for 2000 time steps and frequencies are extracted for 3 components. Grid sizes in number of points, number of frequencies extracted and memory footprint in GB are given in the table 1.

(*) See http://seiscope2.osug.fr.
(**) Results checked at the 81 receivers.

Benchmark	LBen	M1Ben	M2Ben	M3Ben	BBen
Grid size (nb pts)	256x512x512	512x512x512	512x512x512	512x512x1024	512x512x1024
Freq. extracted	2	2	4	2	4
Memory footprint (GB)	15	29	35	55	68

Table 1: Benchmarks configurations with grid size in number of points, number of frequencies extracted for the 3 components and memory footprint in GB.

Results:

1. **Comparison between KNL and BDW for the benchmark LBen:**

For the benchmark LBen the 15 GB memory footprint fits in the MCDRAM. Best results are get with MCDRAM used in flat mode and all the memory binded to MCDRAM (thanks to *numactl --membind=1*).
Kernels are 2.8x to 3.3x faster on a KNL mono-socket than on a BDW dual-socket, and the execution total time (220 seconds on KNL) is 2.4x faster (521 seconds on dual-socket BDW). The difference of speed-up between the kernels and the total runtime mainly comes from I/O (read inputs) that is higher with KNL: 54 seconds instead of 13 seconds, and without that gap the speed-up for the total runtime would have been 2.9. These results are shown in the figure 1.

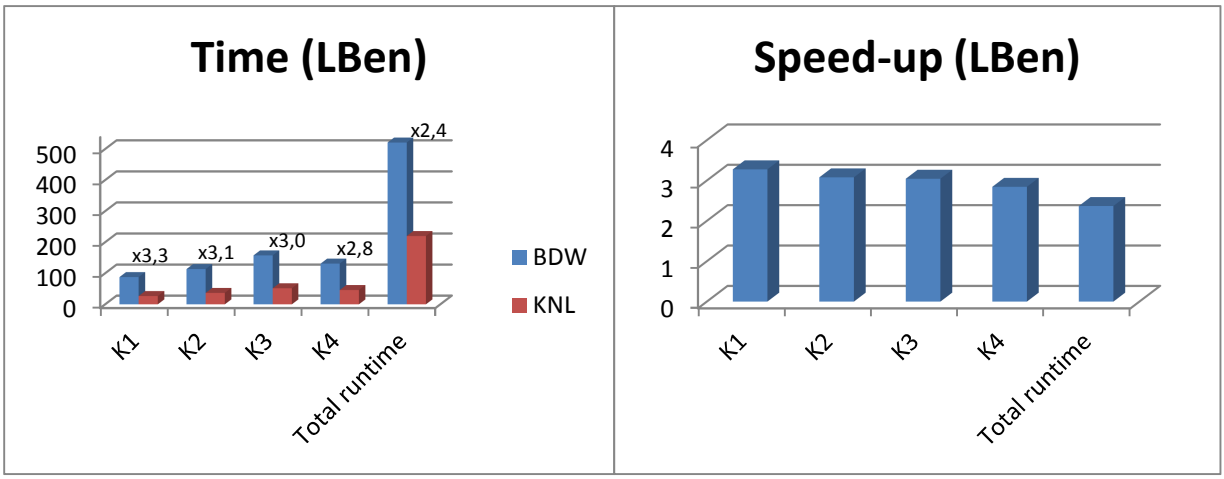

Figure 1: Results for the benchmark LBen. For kernels K1, K2, K3, K4 and total runtime, time in seconds (left) and speed-up (right) for the mono-socket KNL Intel® Xeon Phi™ 7210 compared to the dual-socket BDW Intel® Xeon™ E5-2680 v4.

2. **Comparison of performance functions of the benchmarks:**

To improve efficiency with medium and big benchmarks we have taken care to the thread placement (with variable *KMP_AFFINITY*), we have enabled generation of streaming stores for optimization during compilation (with option - *qopt-streaming-stores=always*), we have activated huge pages and tried different sizes (0 MB, 512 MB, 1048 MB & 2048 MB) for the transparent huge pages of the MCDRAM (thanks to library *tbbmalloc*), and especially, for medium and big benchmarks, we have selected carefully which data are stored on the MCDRAM (thanks to the library *memkind*). However

when memory requirements do not fit anymore in the MCDRAM, performance decreases with the size of the benchmark: see figure 2.

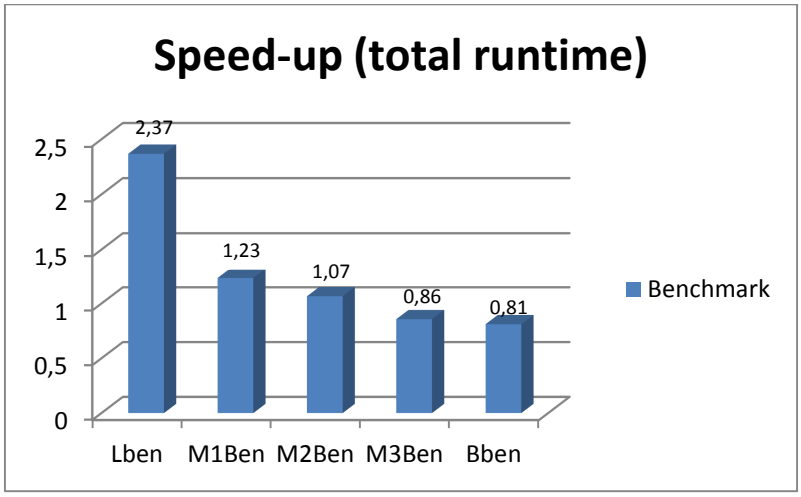

Figure2: Comparison of the speed-up for the mono-socket KNL Intel® Xeon Phi™ 7210 compared to the dual-socket BDW Intel® Xeon™ E5-2680 v4 functions of the benchmark. Performance decreases with the size of the benchmark with values below 1 for 55 and 68 GB benchmarks.

Despite care given to the data stored in the MCDRAM and different methods tried to enable performance, results with the benchmark BBen are far from fully benefiting of the MCDRAM bandwidth: see figure 3.

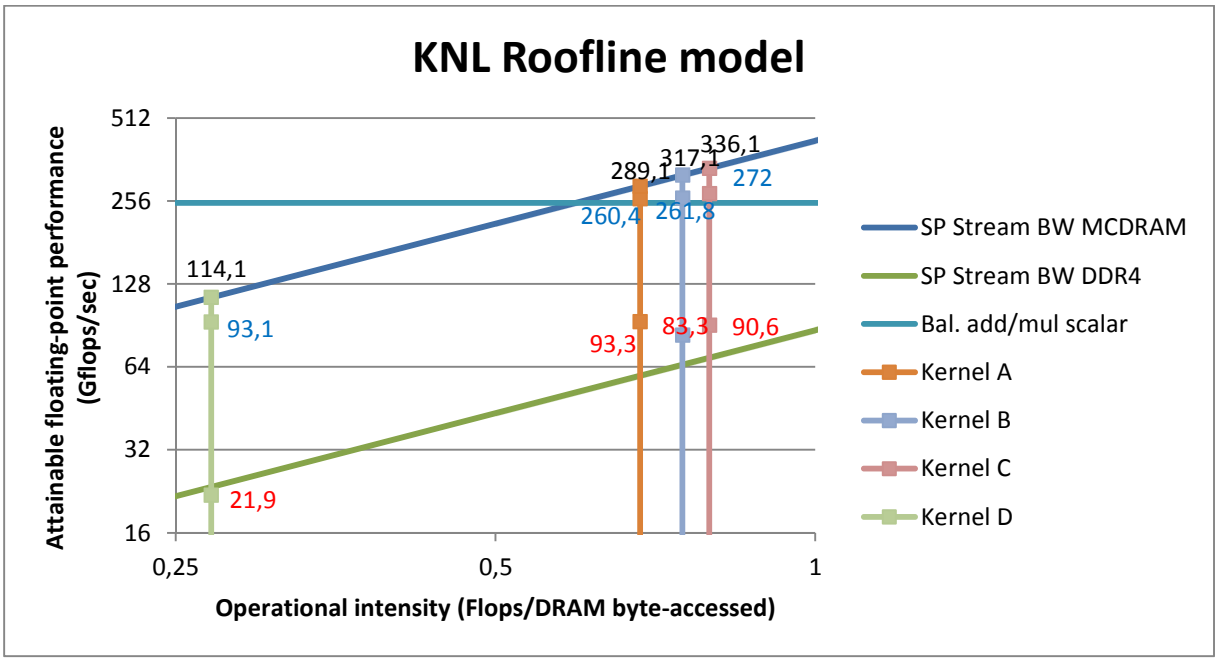

Figure 3: Roofline model for GeoInv3D elastic VTI version on Intel® Xeon Phi™ 7210 with memory using preferentially DDRAM (numactl --preferred=0) but storing 14 critical arrays in the MCDRAM with the memkind library, for benchmarks LBen (blue values) and BBen (red values).

To improve GeoInv3D efficiency, one should consider rewriting its algorithm, and in particular memory accesses, with methods such as vector folding and temporal wave-front tiling that already showed impressive results for some non-staggered stencils.