Geological Processing Performance Analysis on Different Hardware Environments

An Exploration of Effectiveness and Feasibility

Ekaterina Tyutlyaeva, Sergey Konyukhov, Igor Odintsov and Alexander Moskovsky

ZAO "RSC Technologies"

Abstract

In this research, the suitable seismic processing mini-applications were selected with active collaborations with practitioners in seismic data analysis. These mini-applications can serve as a basis for detailed performance study of reverse time migration algorithms, which are actively used in reconstruction of under-surface Earth structure from the seismic sensor readings.

- *The dynamic behavior of chosen mini-applications is studied using the BSC, IPM, MPS performance analysis tools to identify their common features.*
- *Analysis was performed for a set computational architectures, including both common architectures, such as x86, and non-standerde one, vliw.*
- *Performed analysis demonstrates a scalability potential for mini-applications chosen, and we expect more performance /speedup for these mini-application if run on computational cluster in multi-threaded/multi-MPI processes way. That is planned for the future work.*

The main emphasis was put on the computations, while mini-applications I/O requirements, which play important role during data processing and affect total processing time, need to be investigated further.

Algorithms

For performance analysis we have used the most typical seismic mini-application, that implemented 2D and 3D seismic migrations, based on the algorithms used in practice. These applications have been chosen in cooperation with practicing researchers in this field.

At the same time, the applications are characterized by acceptable level of computation complexity, which allows to use different techniques for testing different computational platforms.
The basic flowgraph of the seismic migration is represented on the Fig. 1

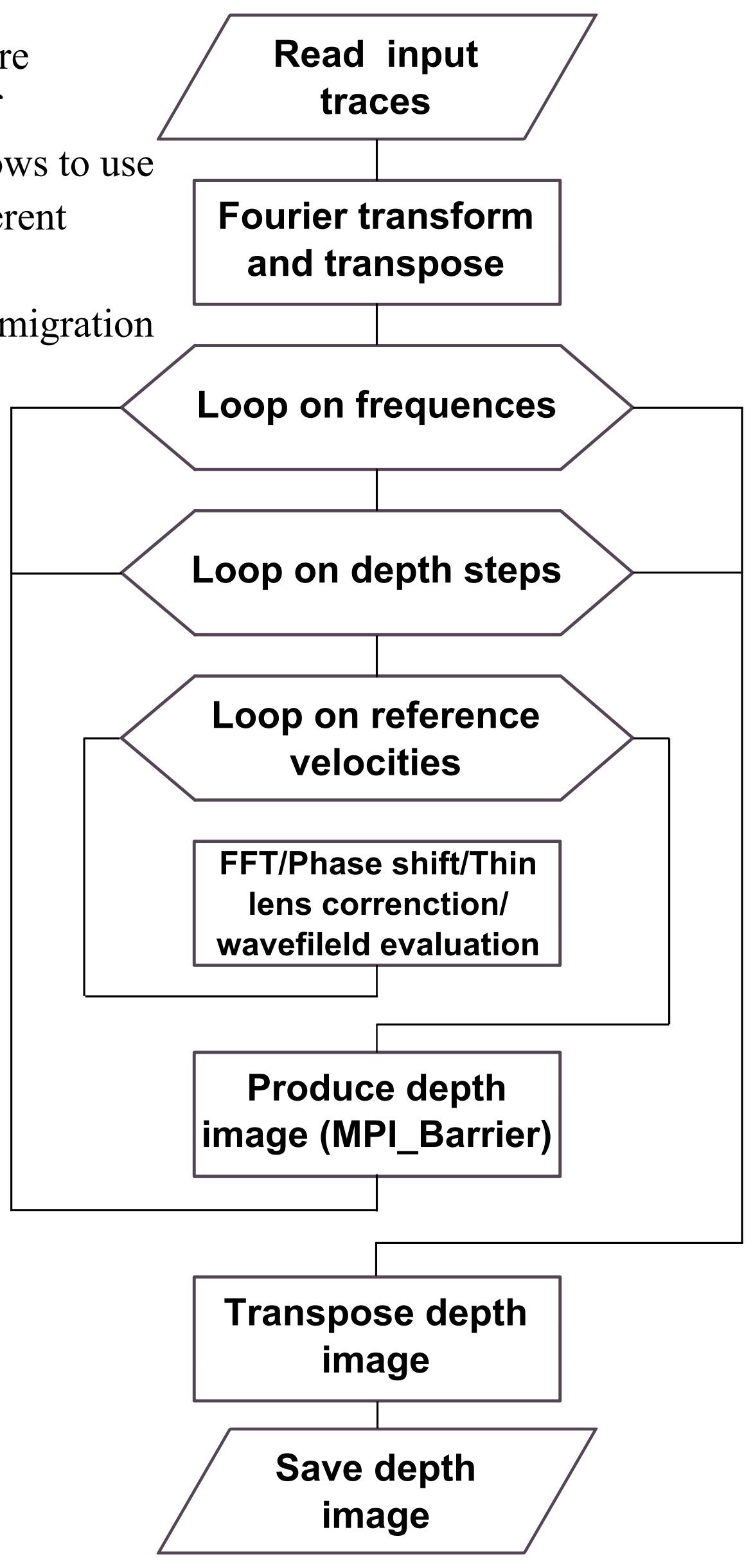
The 2D-seismic migration application (Wemig) uses reverse-time wavefield continuation in frequency/space domains and depth imaging.

Programming model: MPI

The 3D seismic migration application (Cazmig) implements the Cazdag migration algorithm, based on 3D data migration. In this method all computations are performed in the frequency domain where the source and the receiver positions are aligned with the phase shift by the rotation operation of Fourier coefficients.

Programming model: Hybrid MPI+OMP

Figure 1: Seismic migration flowgraph



Hardware

Table 1: Testbeds: Technical Specifications

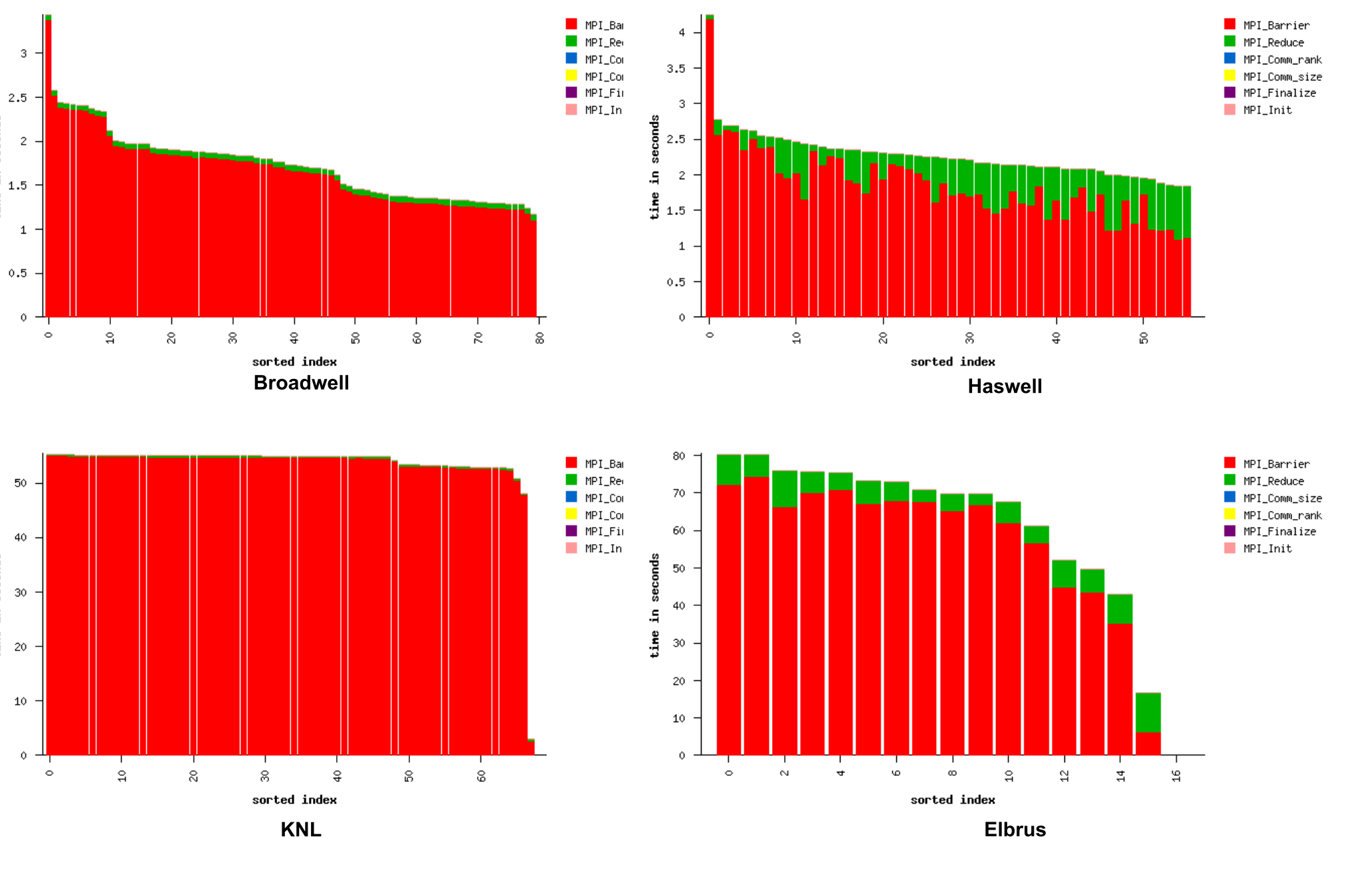
Testbeds	CPU	# Cores	RAM	GB/ core
Haswell_ 64GB	Intel Xeon E5-2697 v3	2x 14	8x DRAM Micron 8GB DDR4/ 2133MHz	2.29
Haswell_ 128GB	Intel Xeon E5-2697 v3	2x 14	8x DRAM Samsung 16GB DDR4/ 2133MHz	4.57
Broadwell_ 64GB	Intel Xeon E5-2698 v4	2x 20	8x DRAM Micron 8GB DDR4/ 2133MHz	1.60
Broadwell_ 128GB	Intel Xeon E5-2698 v4	2x 20	8x DRAM Samsung 16GB DDR4/ 2133MHz	3.20
KNC	Intel Xeon Phi 7120D	61	SDRAM Intel 16GB GDDR5/ 2750MHz	0.26
KNL	Intel Xeon Phi 7250	68	MCDRAM Intel 16GB + 6x DRAM Micron 32GB DDR4/ 2133MHz	2.80
Elbrus	Elbrus-4C	4x 4	12x DRAM Micron 4GB DDR3/ 1600MHz	3.00

Table2: CPU: Technical Specifications

Characteristics	Specifications				
Model	E5-2697 v3	E5-2698 v4	Xeon Phi 7120D	Xeon Phi 7250	Elbrus-4C
Architecture	x86_64	x86_64	x86_64 (MIC)	x86_64 (MIC)	e2k (VLIW)
Clock Speed, GHz	2.6	2.2	1.238	1.400	0.8
Number of Cores	14	20	61	68	4
Number of Threads	28	40	244	272	4
Peak Performance (double precision), GFLOPS	582.4	665.6	2416.6	3046.4	25.6
Max Memory Bandwidth, GB/ sec	68.0	76.8	35.2	76.8	38.4

Profiling: IPM

Figure 2: 2D Seismic Migration Communication Balance by Task (Sorted by MPI time)

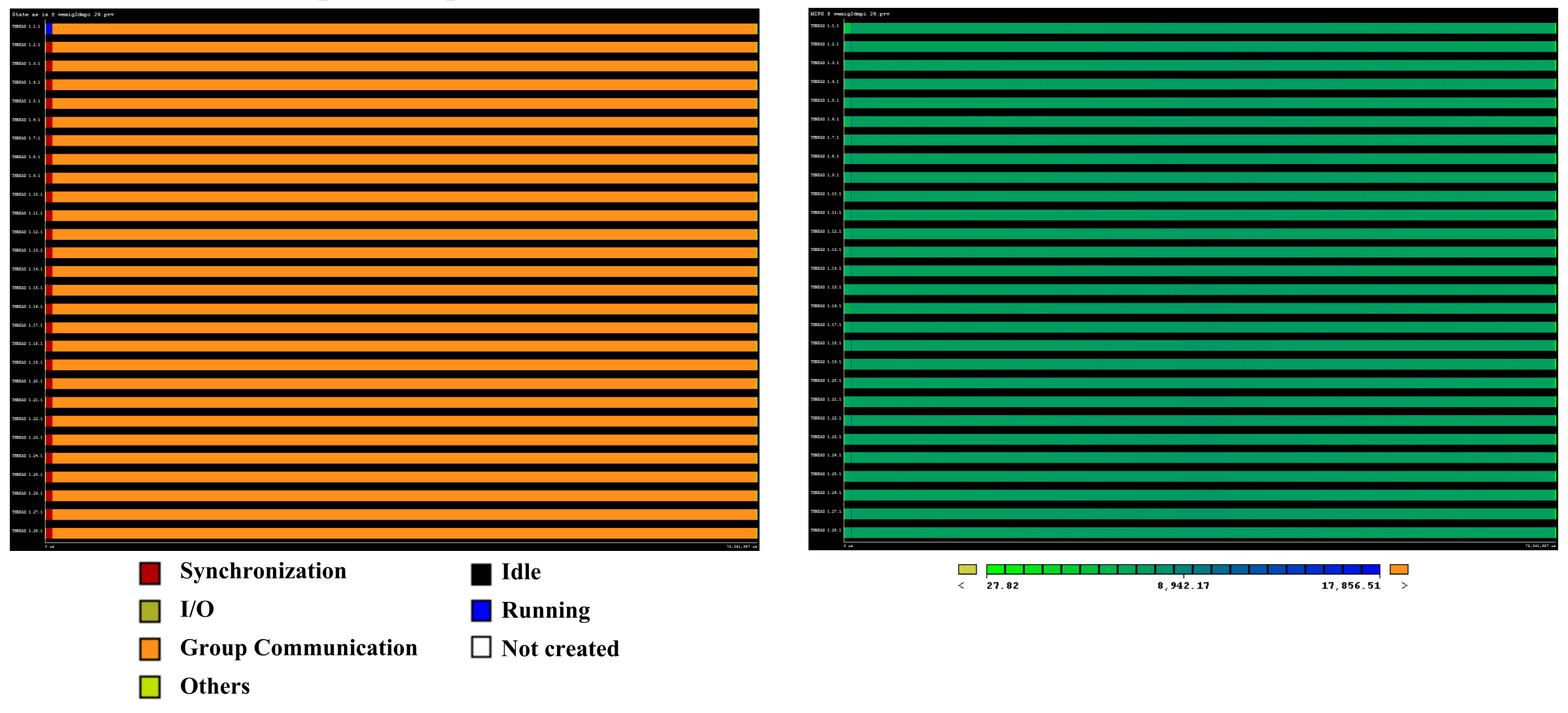


Tracing

2D Seismic Migration Mini-Application Tracing

The analysis of dynamic application behavior based on traces was performed using BSC performance tools.
The application structure is suitable for the parallel execution: the CPU workload is equally distributed during the runtime and MPI ranks, the I/O and synchronization phases are slight. The figure 2 shows the tracing results for the Haswell_128GB architectures, where each horizontal line represents the timelined view of each MPI rank and different computational phases (a) and workload intensity (b) are denoted by different colors;
The tracing results structure for the other studied architectures are similar, while the percentage of computation phases duration and workload intensity differs.

Figure 3: Paraver tracing results for Haswell_128GB

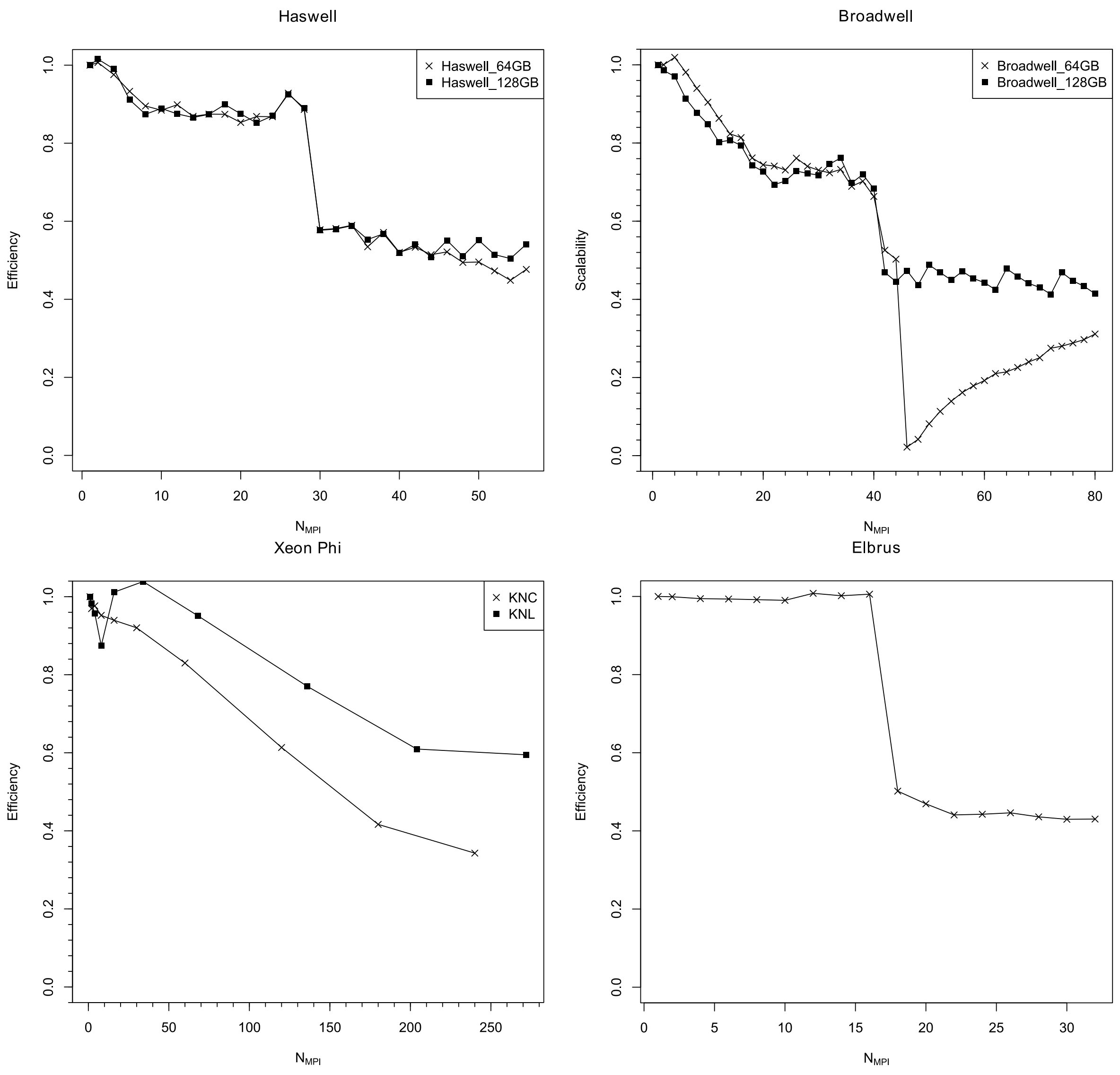

a) Computation phases **b) MIPS**

Parallel Efficiency

Experiments with different numbers of used computation cores have shown the efficiency of the 2D seismic migration mini-app depending on the hardware (see Fig. 3).

$$\text{Efficiency} = \frac{T_{run}(1)}{N \times T_{run}(N)}$$

Figure 4: Parallel Execution Efficiency



It is interesting to note that the doubling of memory capacity leads to the significant performance increase on the Broadwell testbed, while on the Haswell testbed productivity gains are not substantial. It seems the Broadwell cores with 64GB RAM configuration were stalled due to memory demands; the Haswell_64GB results are more balanced. Execution times results presented in Table 6 and Table 7 also validates this observation.
However, the increase in number of MPI processes results in declining efficiency rate for all architectures, despite enabled hyperthreading technology, that provides some performance increase in absolute values.
Comparison of efficiency results for Intel Xeon Phi architecture shows that the second generation architecture (KNL) has significant efficiency and scalability advantage over the first generation (KNC).
Finally, Elbrus-4S architecture demonstrates the almost perfect efficiency up to 16 processes (i.e. up to one MPI process per computational core), because computational complexity is high for this architecture.

Execution behaviour

Table 3: Test Runtimes for 2D Seismic Migration Mini-Application

	Min N_{MPI} time		Max N_{MPI} time	
Testbed	N_{MPI}	T_{run}	N_{MPI}	T_{run}
Haswell_ 64GB	2	12 min 21 sec	56	56 sec
Haswell_ 128GB	2	12 min 15 sec	56	50 sec
Broadwell_ 64GB	2	16 min 24 sec	80	55 sec
Broadwell_ 128GB	2	10 min 38 sec	80	36 sec
KNC	2	550 min 21 sec	240	12 min 58 sec
KNL	2	246 min 34 sec	272	2 min 59 sec
Elbrus	2	280 min 11 sec	16	35 min 47 sec

Table 4: Minimal Execution Time T_{run} for 3D Seismic Migration

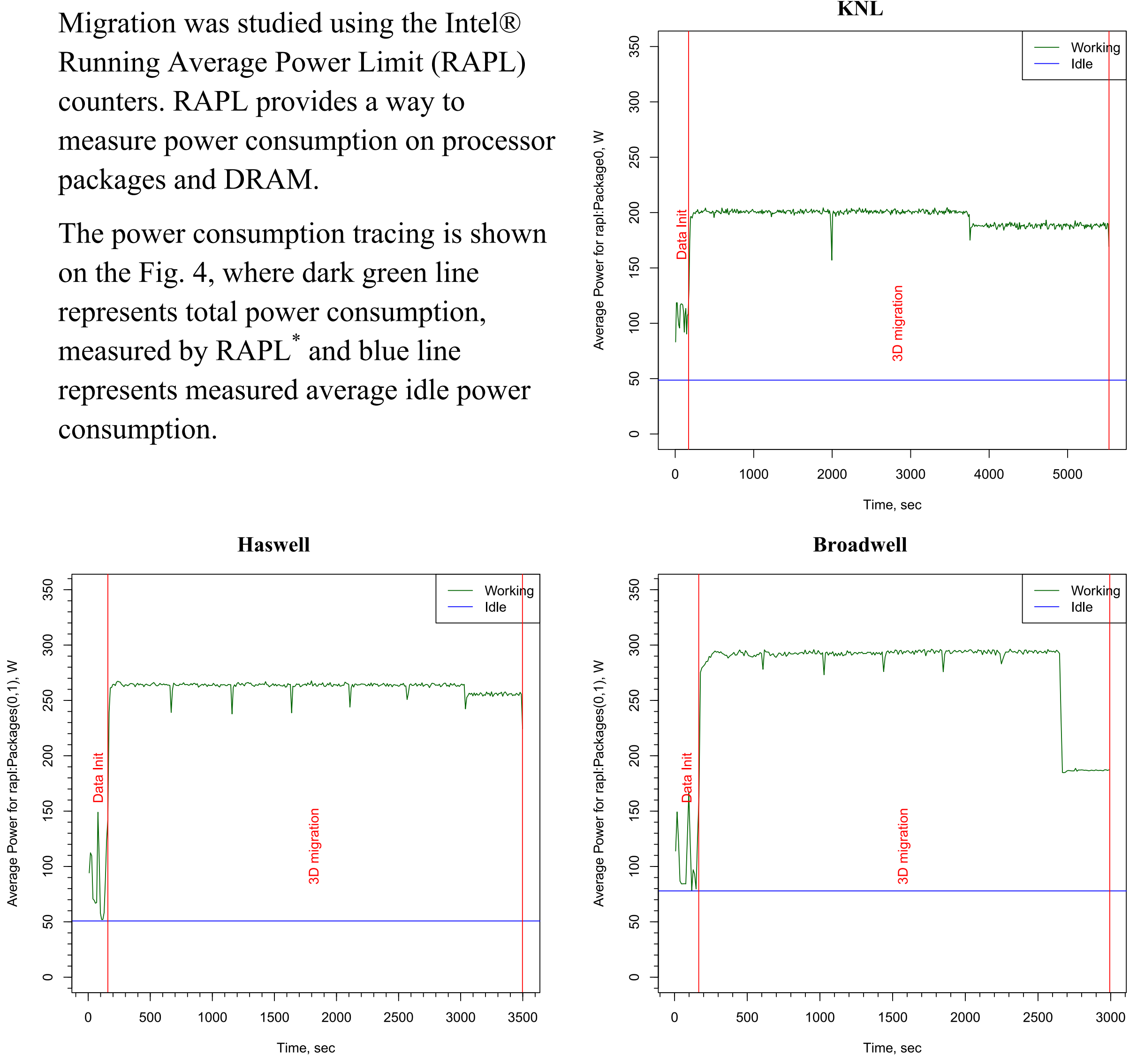
Testbeds	T_{run}	N_{MPI}	N_{OMP}
Haswell_64GB	57 min 35 sec	14	14
Haswell_128GB	55 min 39 sec	14	14
Broadwell_64GB	56 min 13 sec	4	16
Broadwell_128GB	43 min 2 sec	16	16
KNL	70 min 30 sec	34	68
Elbrus	1884 min 42 sec	4	4

Power Consumption

The energy consumption of 3D Seismic Migration was studied using the Intel® Running Average Power Limit (RAPL) counters. RAPL provides a way to measure power consumption on processor packages and DRAM.

The power consumption tracing is shown on the Fig. 4, where dark green line represents total power consumption, measured by RAPL* and blue line represents measured average idle power consumption.

Figure 5: 3D Seismic Migration Power Consumption Tracing



Although during 3D-seismic migration test power consumption rate for Broadwell was higher than for KNL (270 W vs. 188W), total power consumption for Broadwell was lower (0.2245 kW*h vs. 0.3002 kW*h) because of substantially lesser runtime (2988 sec vs. 5526 sec). The Haswell power consumption results show intermediate values (see Table 5).

* *All Packages and DRAM results per node summed up*

Table 5: 3D Seismic Migration Power Consumption

Testbed	Haswell_ 128	Broadwell_ 128	KNL
Energy, kWh	0.2475	0,2245	0.3002

MPI Performance Snapshot

Table 6: 3D seismic migration execution features

Characteristic	Haswell_ 128GB	Broadwell_ 128GB	KNL
Calculation time, %	95.92	94.10	90.52
MPI time, %	4.08	5.90	9.48
OpenMP time, %	86.58	83.13	83.12
I/ O wait, sec	0.53	0.00	903.02
I/ O operations, %	0.00	0.00	0.73

Table 7: The Communication/Computation Ratio (%) for 2D and 3D Seismic Migration Mini-Apps

Application	Haswell_ 128GB	Broadwell_ 128GB	KNL	Elbrus
wemig, %	4.5 / 95.5	10.9 / 89.1	33 / 67	3 / 97
cazmig, %	4.1 / 95.9	5.9 / 94.1	9.5 / 90.5	4.3 / 95.7

Conclusions

In this research we have used computational-intensive software that implements 2D (Cazmig) and 3D (Wemig) seismic migrations to study the application behavior for a set of the computational architectures. In addition to three architecture type comparative analysis, two CPU generation comparisons have been done.

For Haswell/Broadwell testbeds with similar architecture there has been a substantial (about 2x times) performance growth between generations; for the KNC/KNL testbeds the performance increase amounted up to 4x times. Moreover, there is portability issues with KNC architecture that are eliminated in KNL software stack. While the I/O overhead costs are non-essential (0.0% of overall runtime) for most studied architectures, for KNL it takes 0.73% of the runtime. KNC runtime results have worse scalability than the KNL due to lesser amount of RAM per core.

It is worth noting that the doubling of RAM memory capacity leads to the significant performance increase on the Broadwell testbed, while on the Haswell testbed productivity gains are not substantial. So the memory amount for seismic applications should be appropriate to avoid the CPU stalls. The Elbrus-4S CPUs shows the best scalability while overall absolute values were lower than values for the Intel Xeons according to the theoretical performance value rates.

Average power consumption rate is the lowest for KNL and the largest for Broadwell; but total power consumption for 3D seismic migration run shows the best rates for Broadwell testbed.

In summary, it makes sense for seismic applications to use the Intel Xeon E5-2698 CPU generation instead of E5-2697 only with large amount of RAM available; the Intel Xeon Phi 7250 particular architectural characteristics requires careful source code optimizations to help the compiler to effectively vectorize time-consuming loops and to improve the cache locality for achieving higher performance level; The Elbrus-4S CPU is theoretically suitable for such kind of applications, but it requires the frequency and RAM bandwidth increasing, as well as sophisticated source code optimization work.

Acknowledgments
This research was supported by the Common State Scientific and Technological Programme ``SKIF-Nedra'' with funding from Ministry of Education and Science of the Russian Federation.
We thank our colleagues from GEOLAB, particularly Evgeny Kurin for the collaboration.