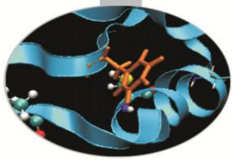
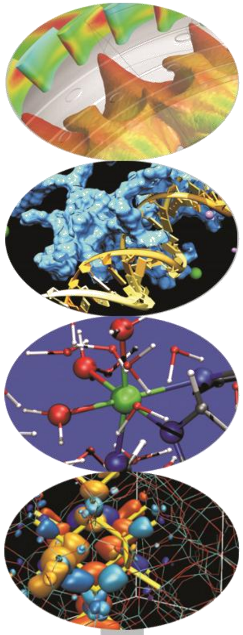
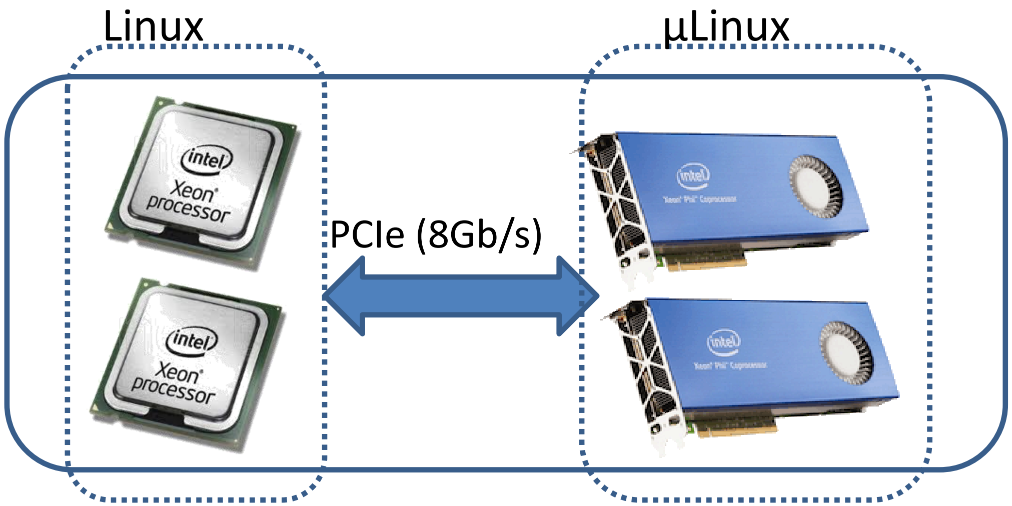


Molecular Dynamics on hybrid-clusters II: Intel Xeon PHI



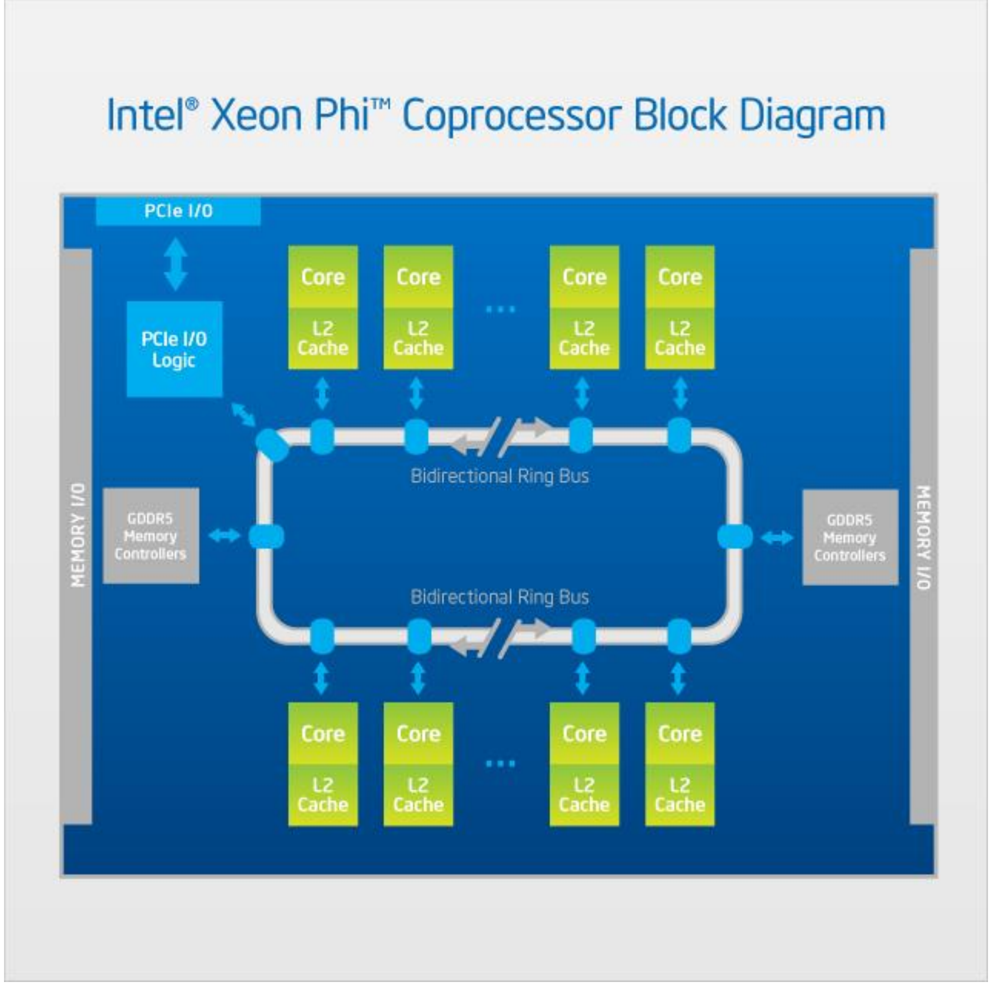
Intel Xeon PHI overview

- Intel product line based on Intel’s Many Integrated Core (MIC) technology where many low, power cores (>50) are packed on a single chip.
- Currently available device (Knight’s Corner or KNC) can be seen as a co-processor, in direct competition to NVIDIA GPU for HPC.
  - connection to host CPU via PCI-eXpress link.
- But unlike GPU technology is not too dissimilar from host CPU → not essential to rewrite code, in principle just re-compile (with Intel compilers). Should lead to shorter development path in most cases.
- Doesn’t mean though that code does need not porting – to obtain peak performance some optimisation is needed.



Most powerful supercomputer in TOP500 (Tianhe-2) uses 48000 Xeon PHI cards.

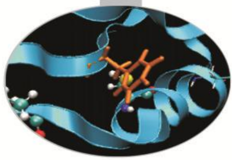
Xeon PHI architecture (KNC)



- 22nm technology
- Up to 61 cores (Pentium-like), ~1.1 Ghz (dep. model)
- 352 Gb/s memory bandwidth (*fast*).
- Upto 244 threads (i.e. 4 threads/core)
- 512 bit SIMD (vector) unit
- 8-16 Gb on board memory,
- ~ 1Tflop peak performance

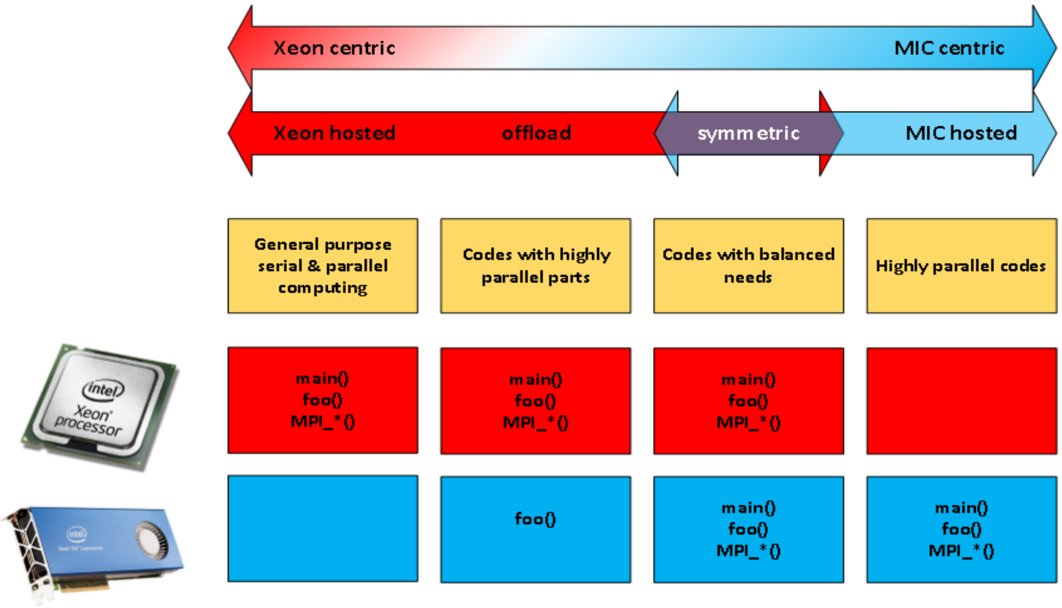
Includes also sensors for monitoring temperature and power consumption.

Xeon PHI software models



Three software models are supported:

1. Co-processor only (native mode): MPI ranks are solely on the Xeon PHI. App can be launched from host or co-processor.
2. Symmetric model: MPI ranks on both host and co-processor.
3. Off-load model: MPI ranks only on host which offload work (e.g. by OpenCL or OpenMP) onto co-processor.

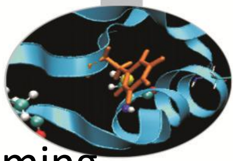


General purpose serial & parallel computing	Codes with highly parallel parts	Codes with balanced needs	Highly parallel codes

main()
foo()
MPI_*()
	main()
	foo()
	MPI_*()
		main()
		foo()
		MPI_*()	

	foo()	main()
		foo()
		MPI_*()
			main()
			foo()
			MPI_*()

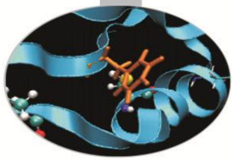
Programming for Xeon PHI



- For the moment must use Intel compilers and Intel MPI for Xeon Phi programming (openmpi planned).
- Off-load model seems to be the most common model for existing HPC applications, by off-loading intensive parts to the XeonPHI (so can re-use CUDA ports).
  - MPI communication in Xeon PHI is quite slow (ring topology). Use shared memory (OpenMP) threads and leave MPI on host CPUs. Can use 4 threads/core.
  - Just like CUDA need to minimise data transfer between host <-> Xeon PHI over slow PCIexpress.
- Intel MKL library ported to Xeon PHI. Many computational chemistry/physics programs “ported” to MIC just by switching to the Xeon PHI MKL.
  - In offload applications MKL can be automatically off-loaded to Xeon PHI
- For peak performance, vectorization must be exploited as much as possible (512 bit SIMD unit) within the Xeon PHI.

```
# environment
module load intel
source $INTEL_HOME/bin/compilervars.sh intel64
# offload code
icc –openmp offload.c  –o offload.exe  # offload code
# MIC native code
icc -openmp –mmic –prog.mic  # MIC native
```

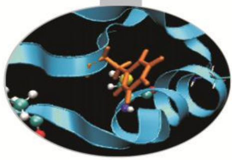
Classical Molecular Dynamics and Intel Xeon PHI



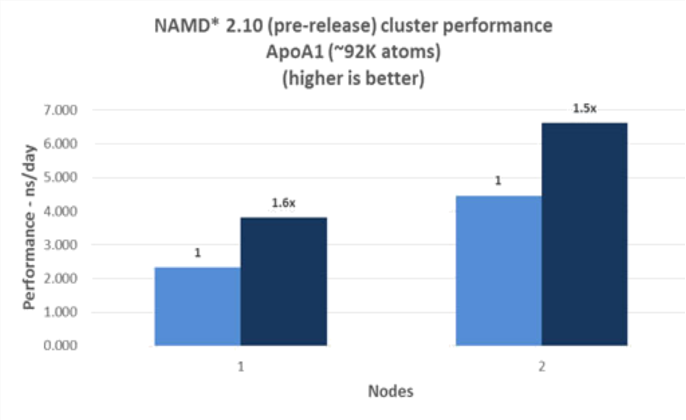
- Development some way behind GPU-CUDA versions of classical MD programs (which started about 4 years ago).
- But given that there is no need to rewrite in new languages (e.g CUDA) development path should be shorter.
- Most current Xeon PHI versions seem to be based on the off-load model to exploit CUDA developments.
- Off-loaded calculations invariably involve non-bonded dispersion interactions may also include PME, energy calculation etc.
- Intel maintains a list of “recipes” for building Xeon PHI applications (not just MD):

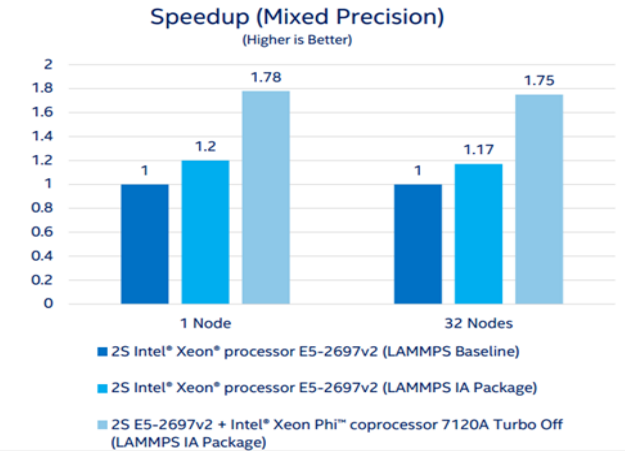
https://software.intel.com/en-us/articles/namd-for-intel-xeon-phi-coprocessor

Molecular Dynamics and accelerators

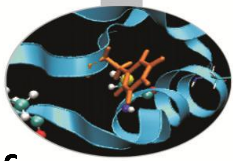


- GROMACS
  - Under development, currently (Jan 2015) only native-mode version available.
- NAMD
  - Pre-release version of NAMD 2.10 has Xeon PHI support but still under development. Speed-ups < 2 for ApoA1 and STMV benchmarks.
- LAMMPS
  - Xeon PHI support available in current downloads for non-bonded calculations. Reported speed-ups of about 1.78x compared to non-accelerated code (one coprocessor/node) for Rhodopsin benchmark. Higher speed-ups obtained with materials simulations.
- AMBER
  - Xeon PHI-enabled version released 6 Aug 2014. Waiting for benchmarks.





Xeon PHI for NAMD

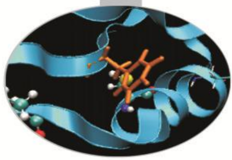


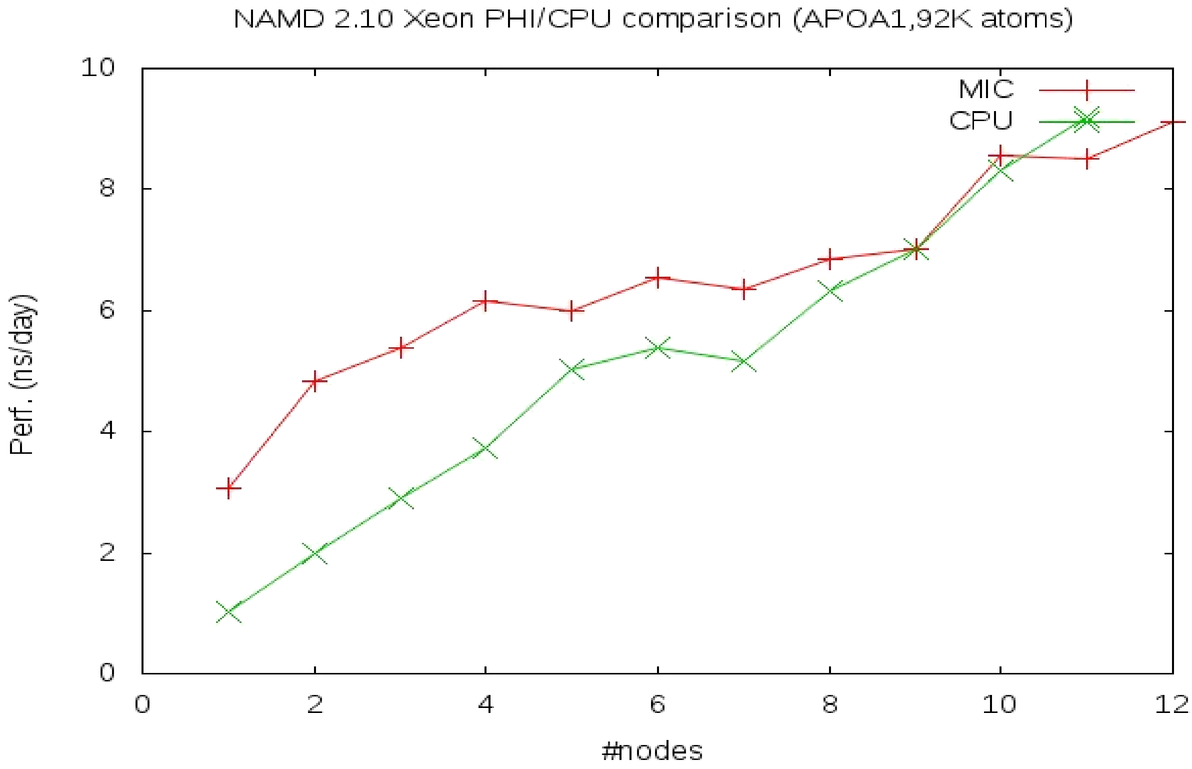
- Uses offload model to calculate non –bonded forces– “clone” of GPU/CUDA offload. MIC tasks overlapped with CPU tasks.
- Charm++ built without MIC support (c.f. GPU)
- NAMD configure with MIC support

```
module load autoload intelmpi   # intel mpi and compilers
# charm++
./build charm++ mpi-linux-x86_64 -with-production

# NAMD configure
./config  Linux-x86_64-icc --charm-arch mpi-linux-x86_64 -with-mic

# running NAMD
source $INTEL_HOME/bin/compilervars.sh intel64
export KMP_AFFINITY=granularity=fine,compact
export MIC_ENV_PREFIX=MIC
export MIC_OMP_NUM_THREADS=240
export MIC_KMP_AFFINITY=granularity=fine,balanced
mpirun -np 32 namd2 md.conf > md.log
```

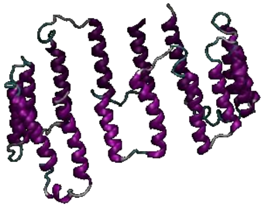
stampede experience https://www.ixpug.org/sites/default/files/Phillips-IXPUG_2014.pdf

Molecular Dynamics and Xeon PHI – early results for NAMD 2.10 on Eurora



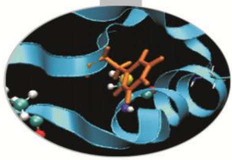


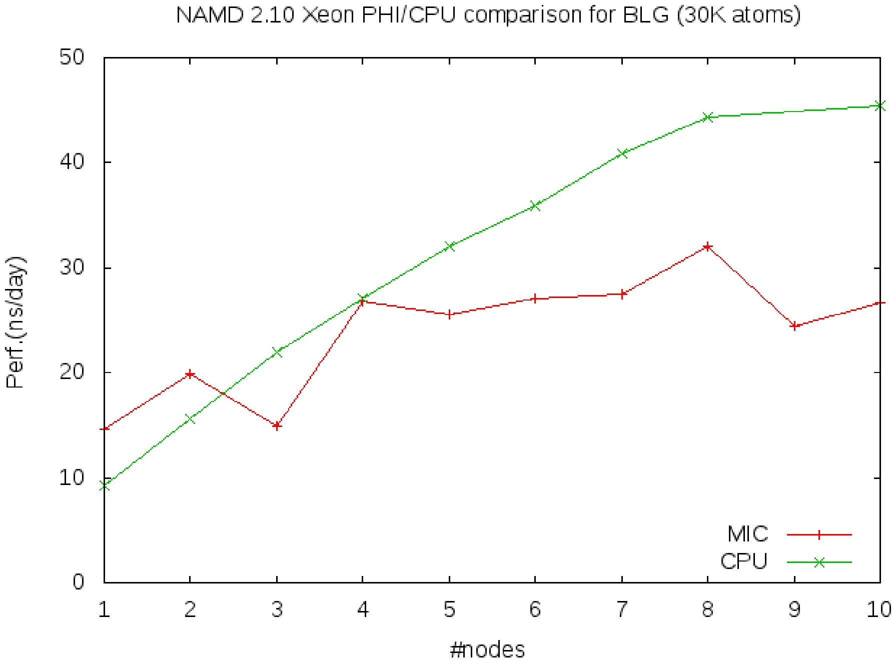
APOA1 92K atoms benchmark



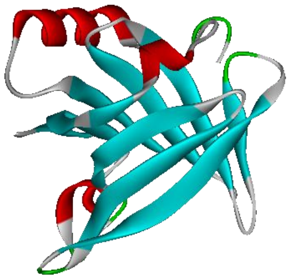
At least for NAMD, Xeon PHI give 2-3x speedups for 1-4 nodes, but the increase isn't maintained.

Molecular Dynamics and Xeon PHI – early results for NAMD on Eurora



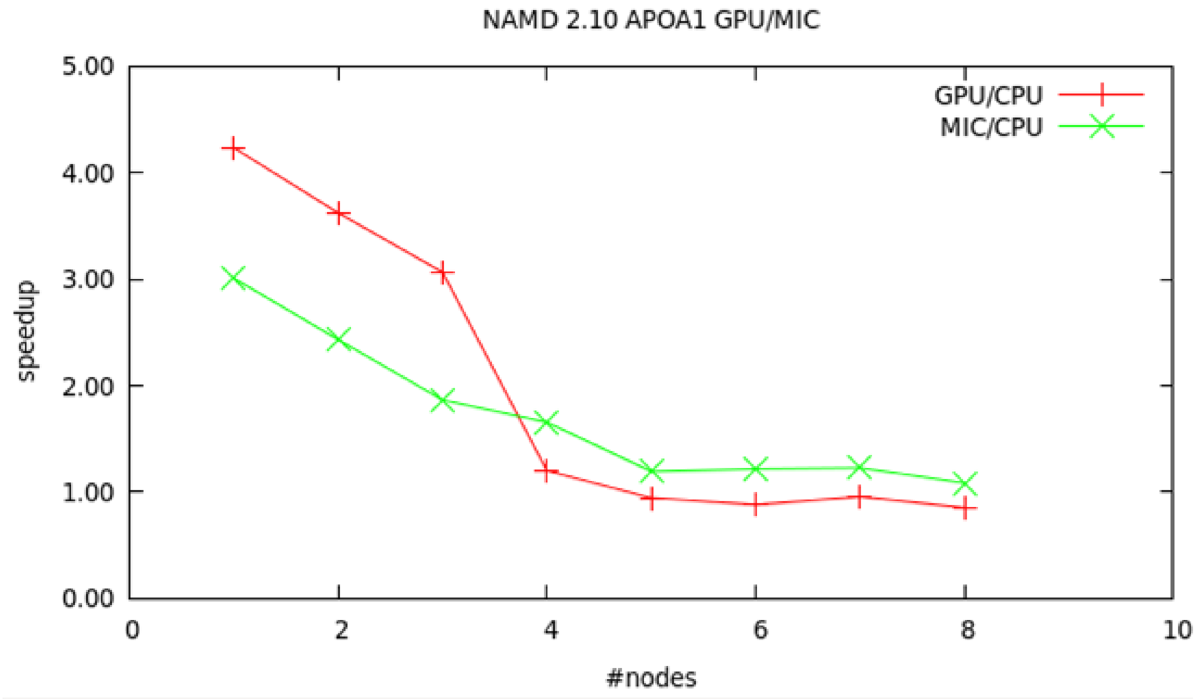


30K atoms



For smaller systems little benefit, so far. Just like GPUs, need reasonable-size system to benefit.

NAMD 2.10 GPU-Xeon PHI speedup comparison (Eurora)

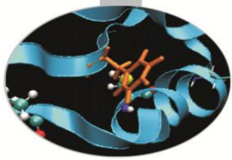


- ..but such curves are misleading because difficult compare equivalent hardware.

- Perhaps Gflops/$ or Gflops/ns would be more realistic.

- In any case the advantage obtained with accelerators, GPU or MIC, disappears at high node counts.

Summary and outlook



- Stable Xeon-PHI accelerated MD codes starting to appear; often implemented in a similar way to CUDA ports (i.e. via the off-load mechanism).
- At least for NAMD, performance broadly similar (perhaps slightly lower) to GPUs although scaling poor. System size important.
- Knight’s Landing processors (KNL) expected second half 2015.
- As well as being more powerful (~3 Tflops), 8-16Gb 3D stacked memory (MCDRAM), etc, will be also available as standalone CPUs.