Partnership For Advanced Computing In Europe

2nd Implementation Phase
Work Package 11

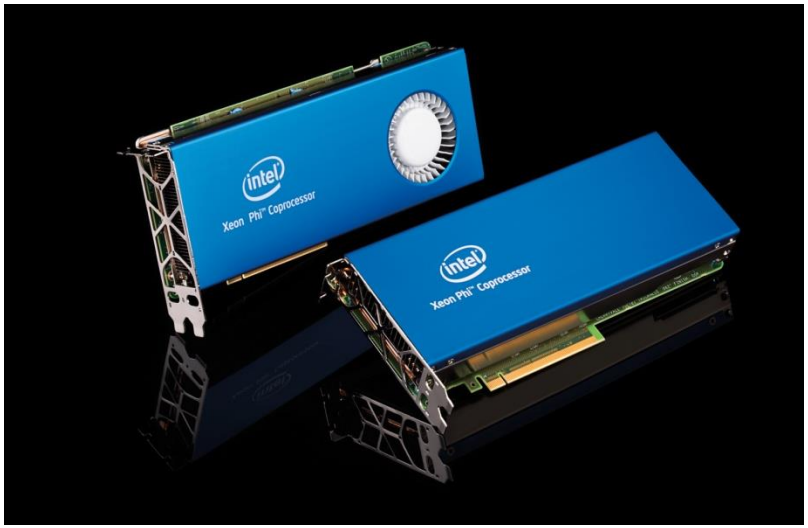
Giannos Stylianou
Research Assistant

The Cyprus Institute

CaSToRC

Intel Xeon Phi Co-Processor 5110P

- 60 cores
- 240 threads
- 1.053 GHz
- 512-bit wide registers
- 8GB RAM
- 30MB Cache Memory
- Max. TDP 225W
- TFLOPs Peak DP Performance



Wilson – Dirac Equation

$$\frac{1}{2\kappa}\psi(x) + \frac{1}{2}\sum_{\mu=0}^{3}(1-\gamma_\mu)u_\mu(x)\psi(x+\hat{\mu}) + (1+\gamma_\mu)u_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu})$$

- QCD applications spend 70-80% of the execution time in this equation
- HEAT EQUATION mimics the hopping in the Wilson-Dirac operator

$$\frac{1}{2\kappa}\psi(x) + \frac{1}{2}\sum_{\mu=0}^{3}(1-\gamma_\mu)u_\mu(x)\psi(x+\hat{\mu}) + (1+\gamma_\mu)u_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu})$$

- SU3 MULTIPLICATION is the piece of the most floating point operations in Wilson-Dirac operator

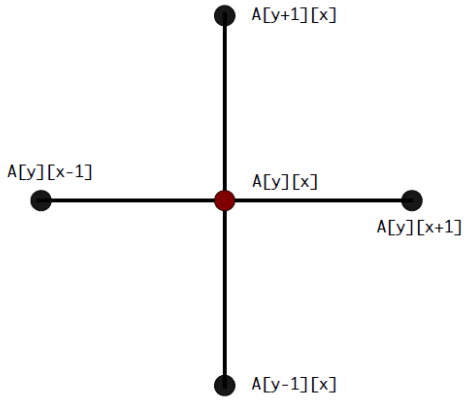
$$\frac{1}{2\kappa}\psi(x) + \frac{1}{2}\sum_{\mu=0}^{3}(1-\gamma_\mu)u_\mu(x)\psi(x+\hat{\mu}) + (1+\gamma_\mu)u_\mu^\dagger(x-\hat{\mu})\psi(x-\hat{\mu})$$

- Both benchmarks are run on Intel Xeon Phi in native mode

Heat Equation Benchmark (1/4)

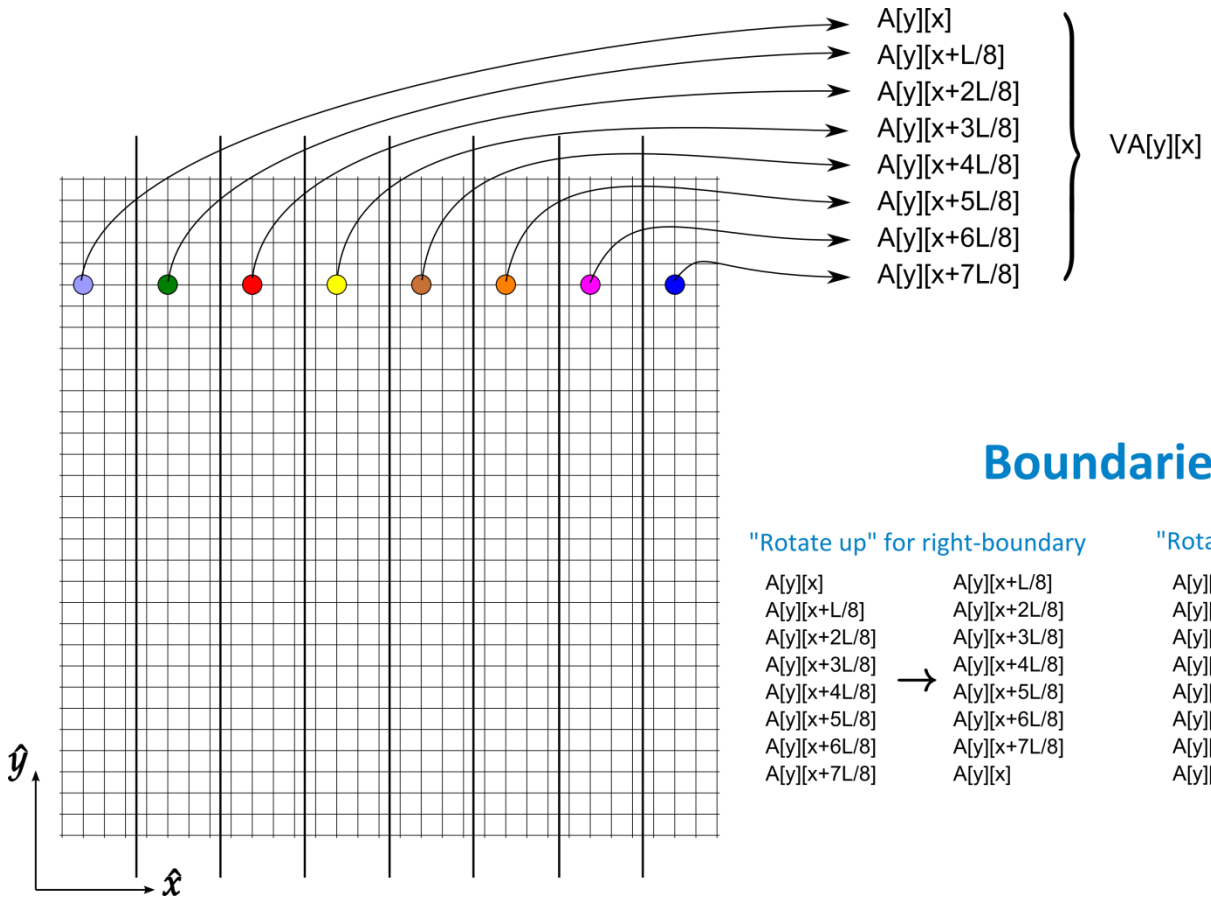
- **Discrete Laplacian Operator in 2D**

$$\frac{1}{1+4\sigma}\left\{\phi(x)+\sigma\left[\phi(x+\hat{i})+\phi(x-\hat{i})+\phi(x+\hat{j})+\phi(x-\hat{j})\right]\right\}$$




```
B[y+0][x+0] = a*A[y+0][x+0] + b*(A[y+0][x+1] + A[y+0][x-1] + A[y+1][x+0] + A[y-1][x+0])
B[y+0][x+1] = a*A[y+0][x+1] + b*(A[y+0][x+2] + A[y+0][x+0] + A[y+1][x+1] + A[y-1][x+1])
B[y+0][x+2] = a*A[y+0][x+2] + b*(A[y+0][x+3] + A[y+0][x+1] + A[y+1][x+2] + A[y-1][x+2])
B[y+0][x+3] = a*A[y+0][x+3] + b*(A[y+0][x+4] + A[y+0][x+2] + A[y+1][x+3] + A[y-1][x+3])
```

Heat Equation Benchmark (2/4)



Boundaries:

"Rotate up" for right-boundary

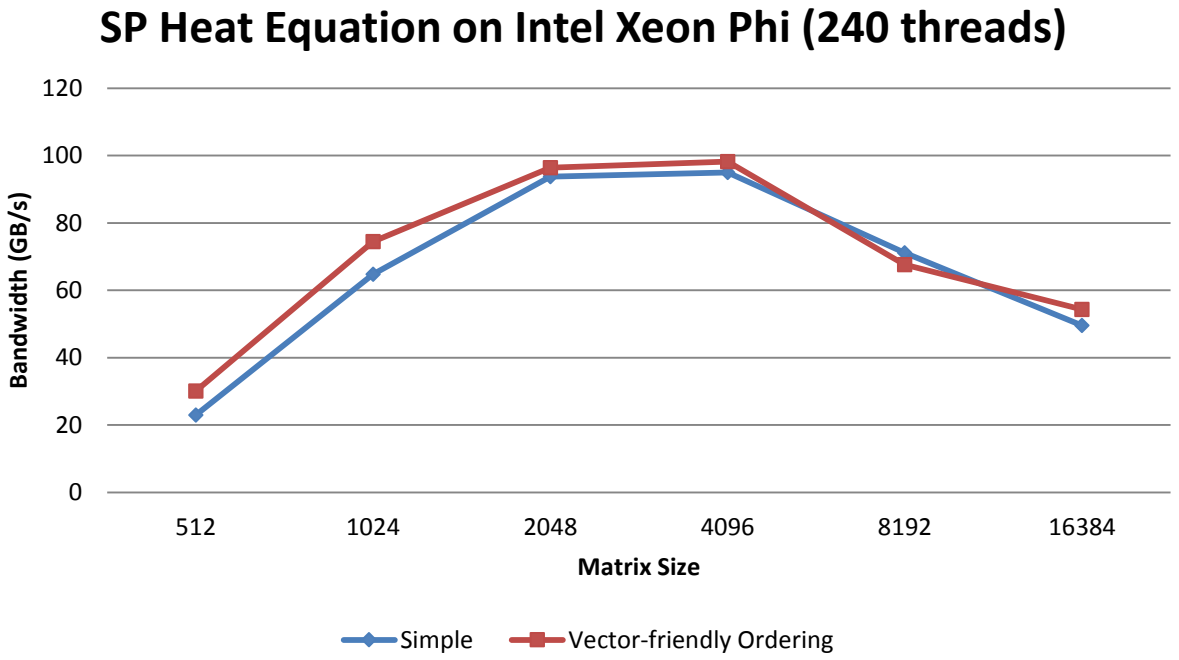
A[y][x]	A[y][x+L/8]
A[y][x+L/8]	A[y][x+2L/8]
A[y][x+2L/8]	A[y][x+3L/8]
A[y][x+3L/8]	A[y][x+4L/8]
A[y][x+4L/8]	A[y][x+5L/8]
A[y][x+5L/8]	A[y][x+6L/8]
A[y][x+6L/8]	A[y][x+7L/8]
A[y][x+7L/8]	A[y][x]

"Rotate down" for left-boundary

A[y][x]	A[y][x+7L/8]
A[y][x+L/8]	A[y][x]
A[y][x+2L/8]	A[y][x+L/8]
A[y][x+3L/8]	A[y][x+2L/8]
A[y][x+4L/8]	A[y][x+3L/8]
A[y][x+5L/8]	A[y][x+4L/8]
A[y][x+6L/8]	A[y][x+5L/8]
A[y][x+7L/8]	A[y][x+6L/8]

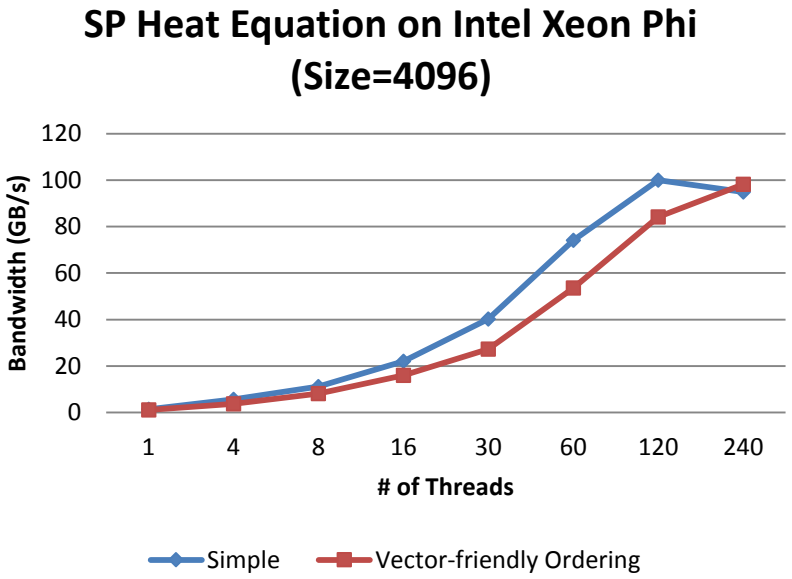
- Vectorization (Double Precision)

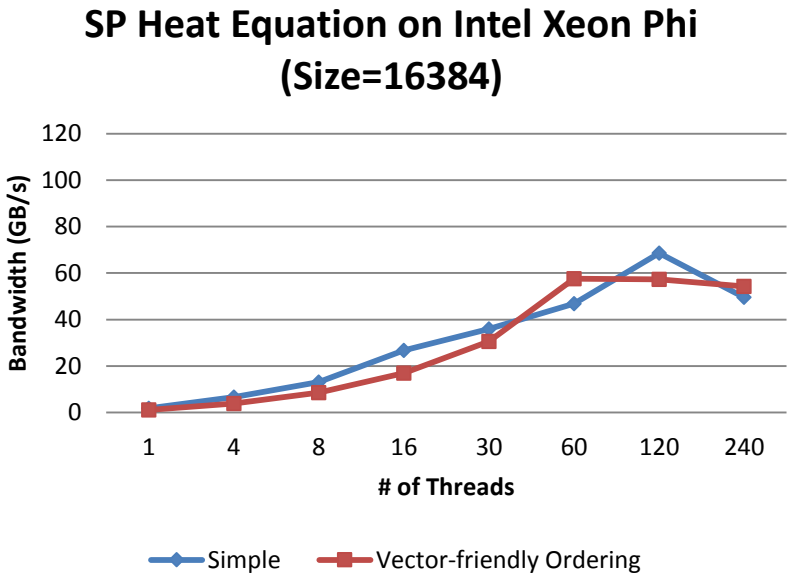
Heat Equation Benchmark (3/4)



- Bandwidth was measured by dividing the total memory I/O with elapsed time.
- Not a significant difference between vectorized and non-vectorized version
- Vector-friendly Ordering is slightly better

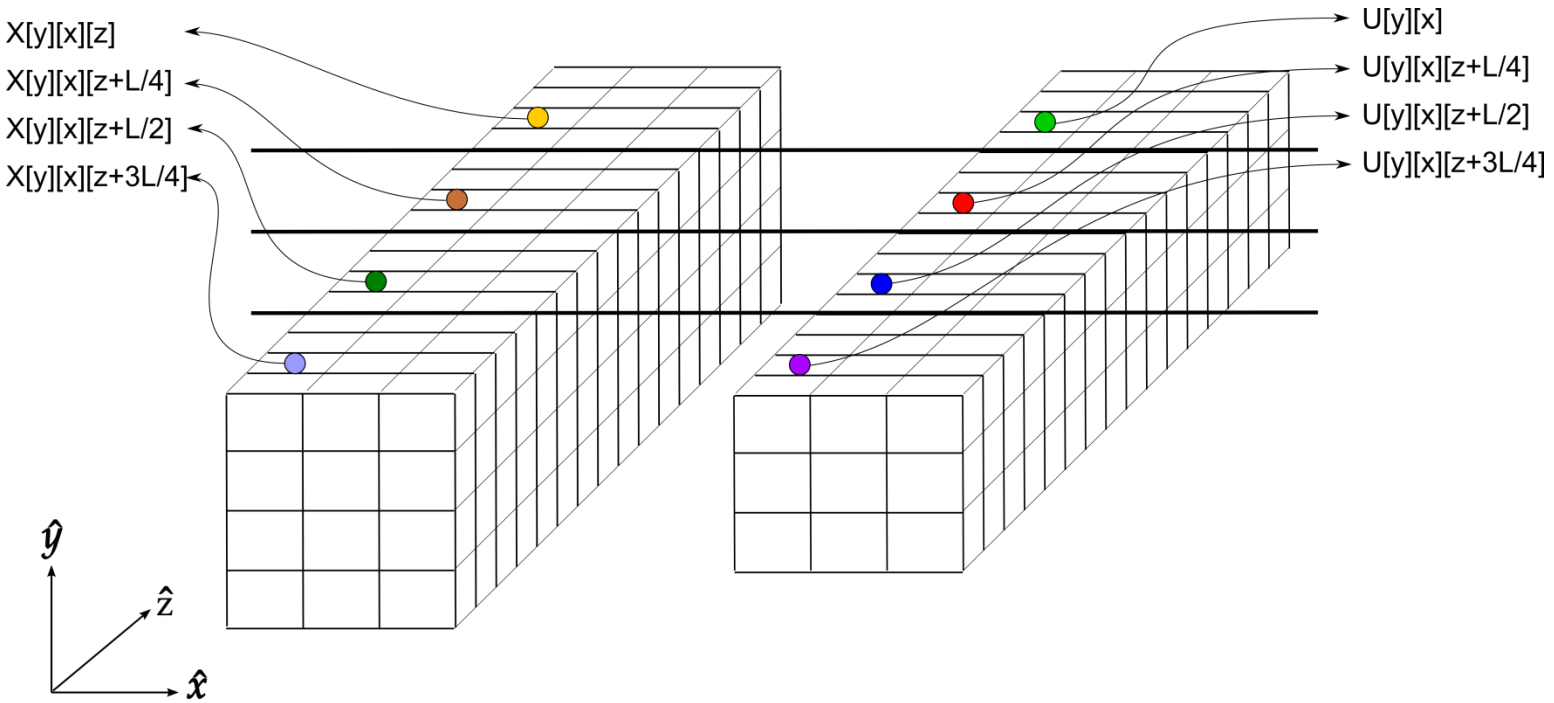
Heat Equation Benchmark (4/4)





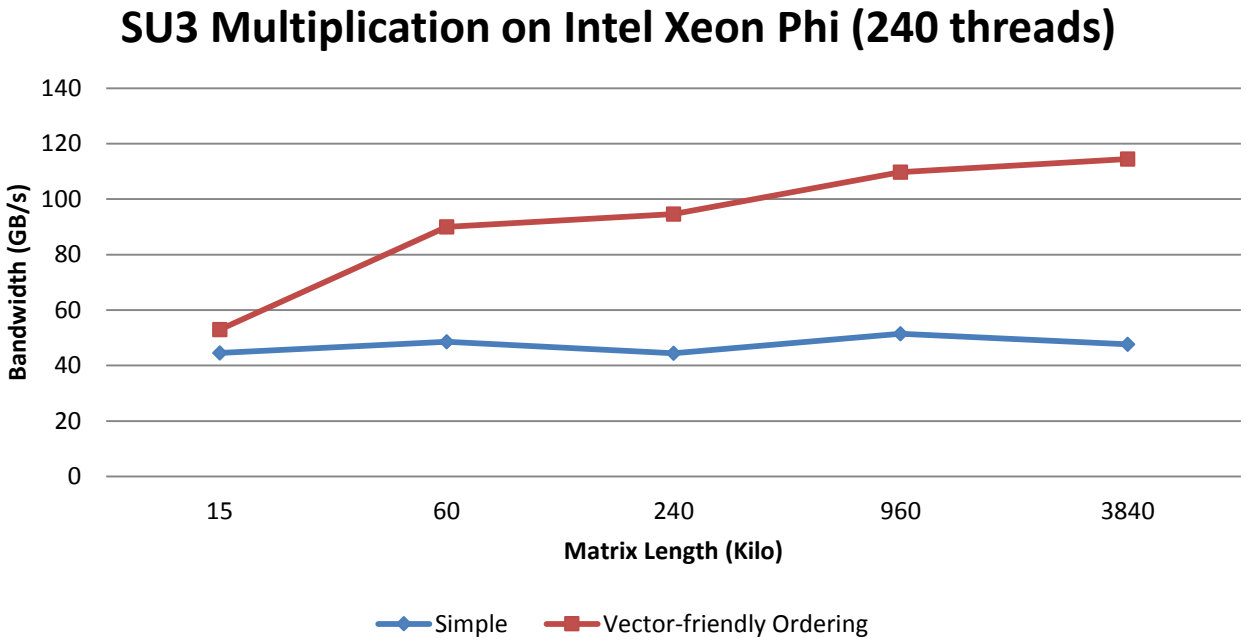
- For small sizes, better to use all the available threads
- For larger sizes, using all the available threads makes no difference
- Achieves a bandwidth of 100 GB/s when the size is not too large
- Conclusion: Compiler auto-vectorization seems to be working well for real numbers

SU3 Mult. Benchmark (1/3)



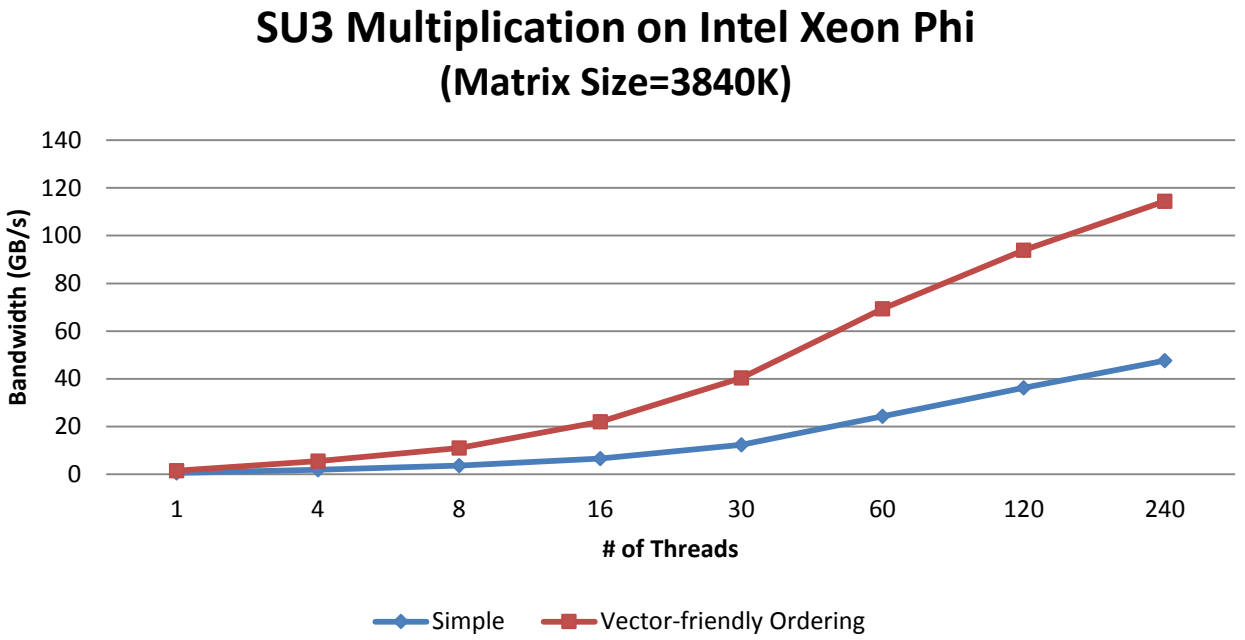
- Vectorization (Double Precision)
- Shuffles required only between real and imaginary parts of the complex numbers
- This ordering will benefit some LQCD applications which are under development

SU3 Mult. Benchmark (2/3)



- Compiler auto-vectorization seems to not like complex multiplication numbers
- Hand-vectorization more than doubles the bandwidth
- Data reordering helped in reducing the hand-vectorization effort
- Constraint: Matrix length must be a multiple of 4

SU3 Mult. Benchmark (3/3)



- Multithreading helps in hiding memory latencies (bandwidth is doubled)
- Vector-friendly ordering scales better
- Achieves a bandwidth of 114 GB/s
- Can sustain over one third of the MIC peak bandwidth with moderate porting effort

Heat Equation & SU3 Multiplication

- ***Next Objective:***

  Integrate these two kernels into the full Wilson-Dirac Equation and run it on the prototypes.

Thank you

www.cyi.ac.cy/castorc

CaSToRC