Hartree Centre

Science & Technology Facilities Council

Luke Mason

Intel® Parallel Computing Center at Hartree Centre, STFC

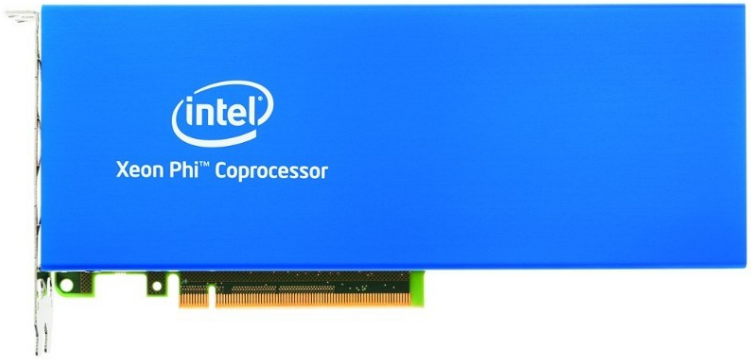
Optimising DL_MESO for Intel Xeon Phi

IPCC at Hartree Centre, STFC

- Intel funded code porting and modernisation project.
- In our second year.
- Intel's many core architectures (Xeon Phi) are main focus.
  - Knights Corner (KnC) is current generation.
  - Knights Landing (KnL) is the next generation.
- Wide variety of codes.
  - DL_MESO.
  - Unified Model.
  - GungHo.
  - DualSPHysics.
  - ParaFEM.
  - OSPRay.

Knights Corner: System Specs

- Coprocessor.
  - **PCI bus.**
- C/C++/Fortran.
- MPI/OpenMP.
- Linux operating system.
- 6 – 16 GB cached GDDR5 RAM.
- 57 – 61 cores.
- Clock speed ~1GHZ.
- 4 Hardware threads per core.
- 512 bit vector unit.
- IMCI instruction set.

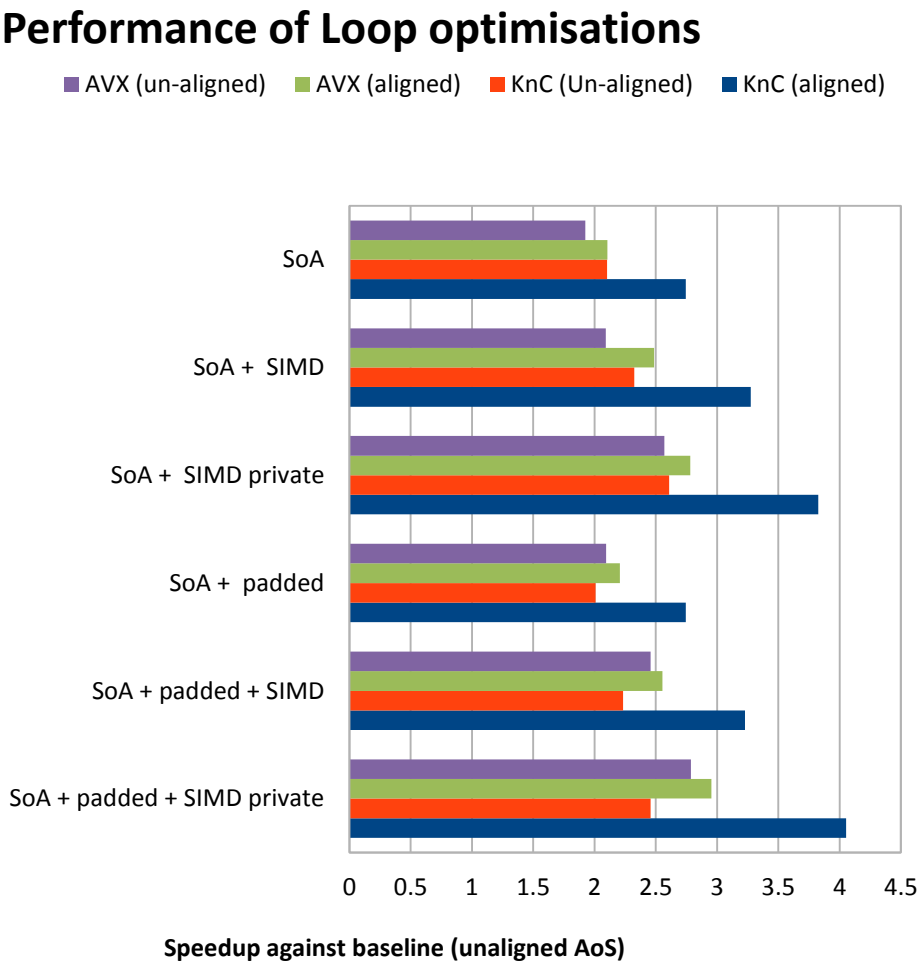


DL_MESO_LBE: Lattice Boltzmann

- Models fluids at mesoscale.
- Hybrid MPI/OpenMP running natively.
- Using compiler 16.0.042.
- Extension of Xeon tuning work performed with Vector Advisor.
- fGetEquilibriumF identified as hotspot
- Baseline characteristics
  - **Double precision**
  - **Array of Structures (AoS) data storage.**
  - **No data alignment (prevented by data structure).**
  - **Low trip count loop (19).**
  - **Trip count is not a multiple of vector lengths.**
  - **Both Peel and remainder loops present.**

DL_MESO_LBE: Enhanced Vectorization

- Advisors recommendations
  - **MAP analysis points to AoS -> SoA.**
  - **Pad arrays to avoid scalar remainders.**
  - **Align data accesses.**
- AoS -> SoA allowed aligned access removing peel loops.
- Array padding and #pragma loop count removes remainder loops.
- Additional optimization
  - **#pragma simd private**
  - **Allows additional compiler optimizations.**
- Final loop speed up of 4.05 over base line performance.
- The same optimizations produced 2.95 speed up on Xeon

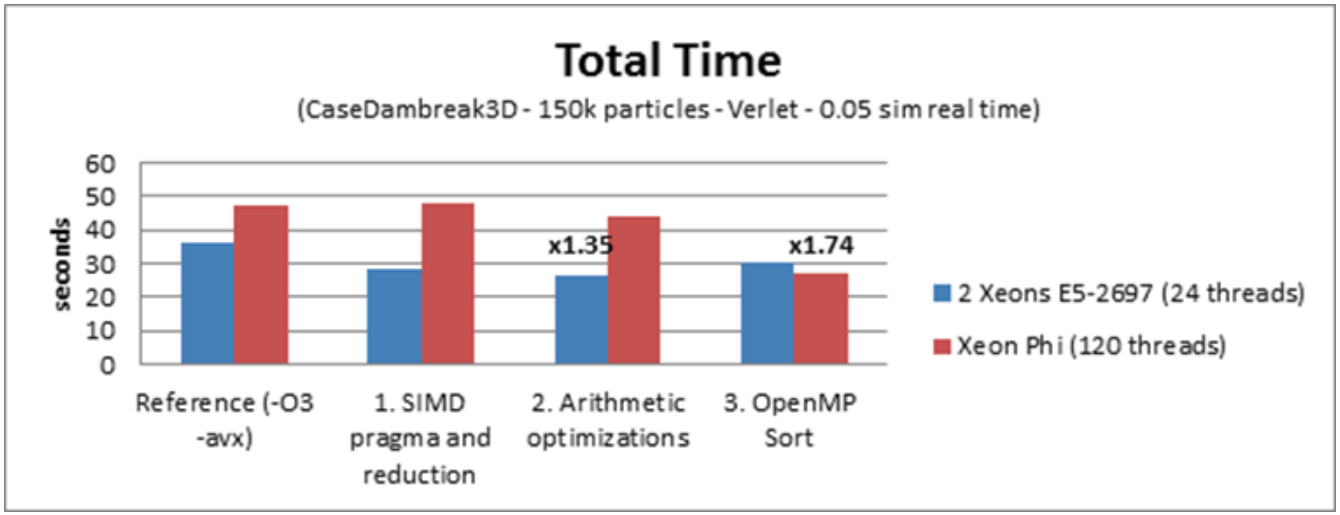


DL_MESO_LBE: Vectorization Insights

- AoS -> SoA transformation gave large gains on KnC, more so than on Xeon.
- Data alignment much more significant for KnC.
- In isolation array padding does not provide significant benefits on KnC.
  - **Compiler has caught us up!**
  - **Remainder loop auto vectorization seems pretty effective.**
  - **This is not true on Xeon where manual padding is still required.**
- When combined with simd private clauses and aligned access manual padding becomes worthwhile however.

DualSPHysics: Parallel Particle Sort

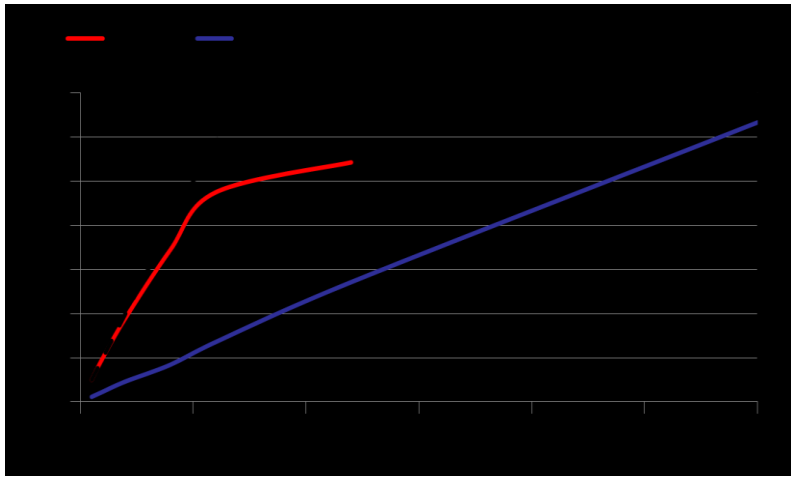
- Optimisations applied.
  - **SIMD pragma + reduction.**
  - **Arithmetic refinement.**
  - **OpenMP sort.**
- Current KnC solver shows equivalent performance to 2 E5-2697 Xeon CPUs.
- Still not fully optimised for KnC.



ParaFEM: MKL

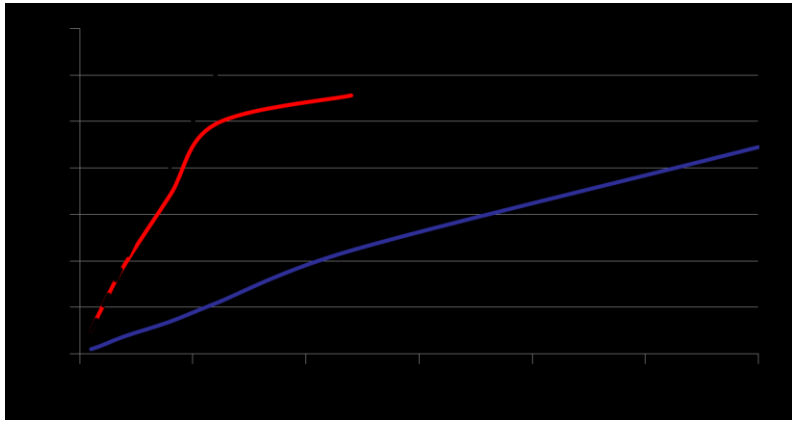
- Exploring native performance on Xeon Phi
  - **Hotspot is loop of matrix vector multiplications**
  - **The matrices are small 60x60**
- Modifications to code so far:
  - **Trialed different methods for the matrix-vector multiplies.**
  - **MKL Sequential BLAS was best.**
  - **Data alignment.**
  - **Streaming stores.**

Hotspot scaling compared to 1 MPI rank on Xeon



MPI Ranks

Solver scaling compared to 1 MPI rank on Xeon



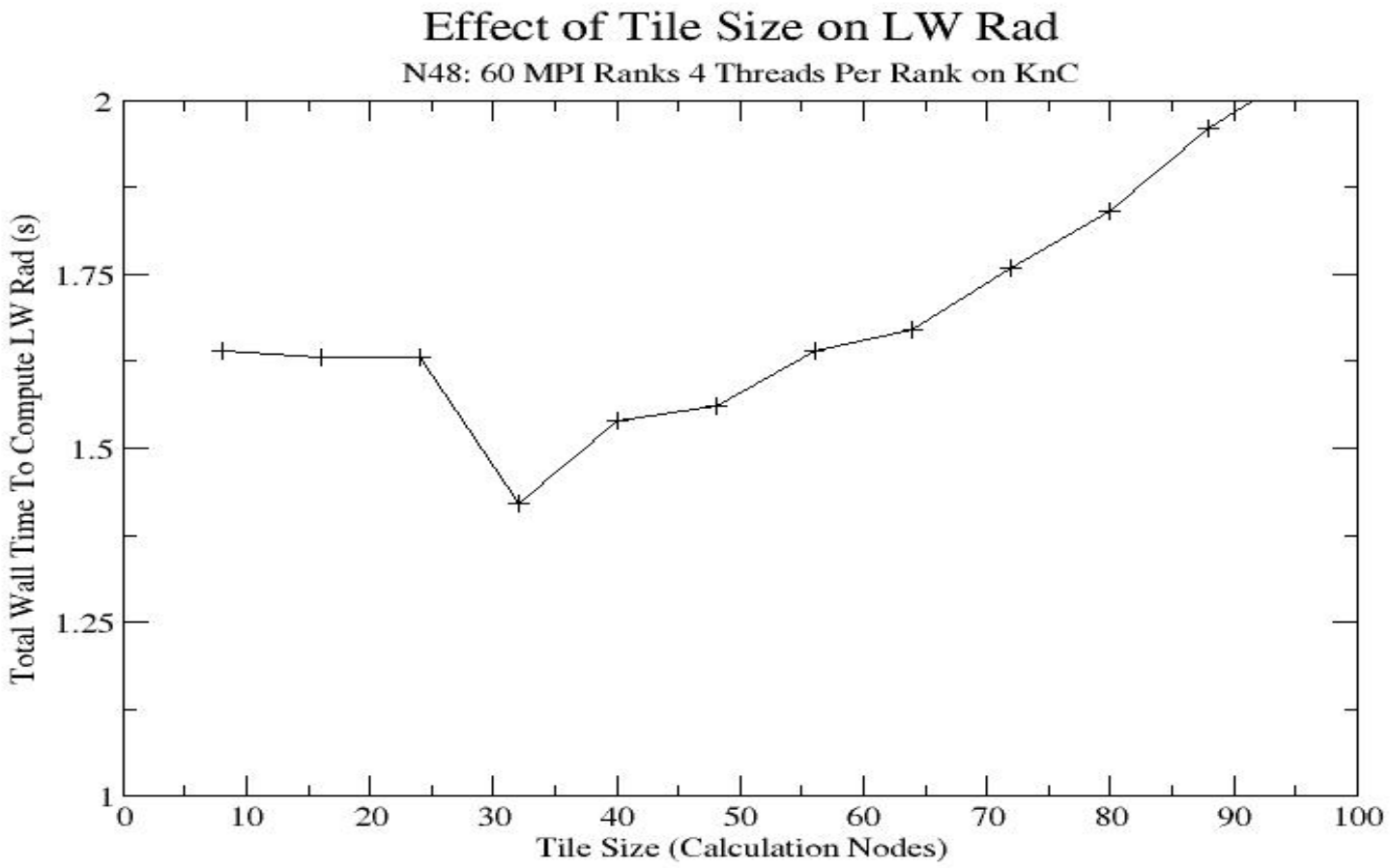
MPI Ranks

Unified Model: Cache Optimisation

- Calculation of Radiation performed at each grid point.
- Total grid points split into tiles.
- Additional outer loop over tiles required.
- Optimal tile size will depend on architecture.
- Solution time can be very sensitive to tile sizes.
- Aim is to provide optimal cache re-usage for the given problem.
- Tiling can aid vectorization.
  - **Set tiling size to a multiple of vector width.**

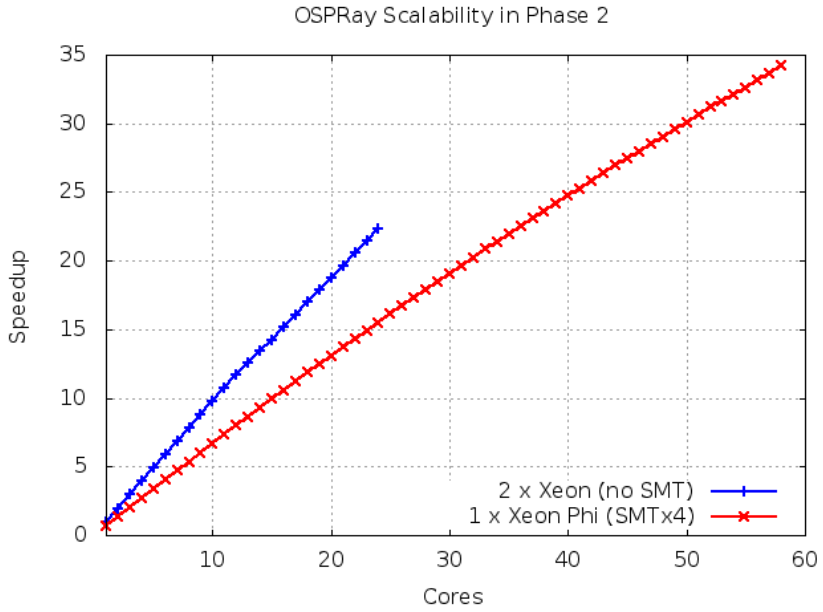
Met Office

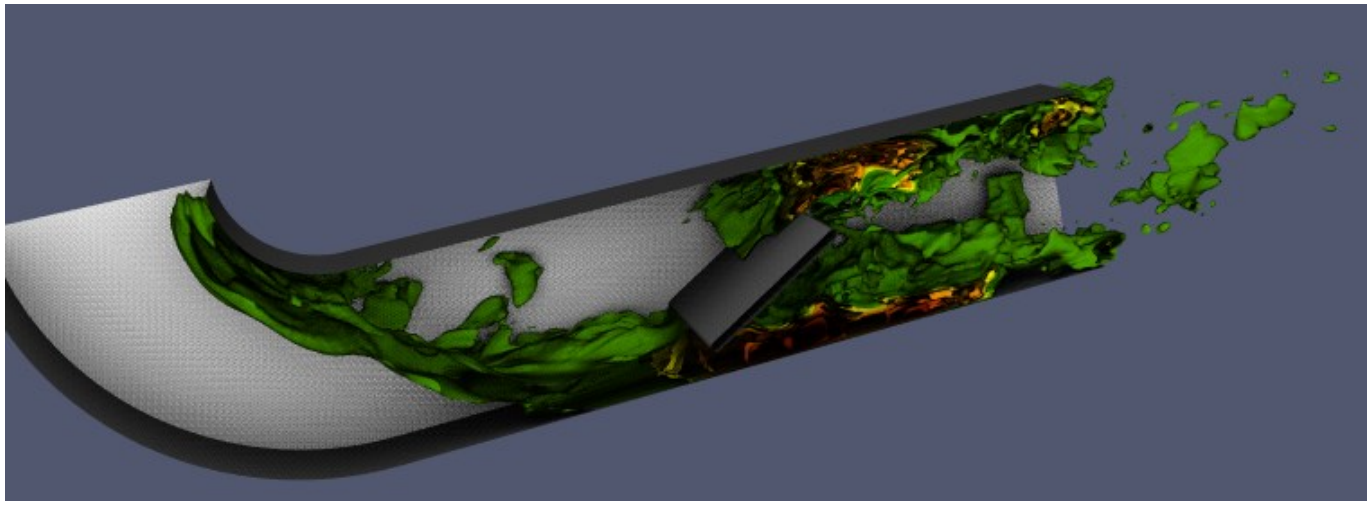
Unified Model: Tuning Results



OSPRay: Ray tracing without GPU.

- Ray tracing built using Embree.
- Open source.
- Plugin for Paraview.
- Usable on CPUs and MICs (KnC).
- Integrated with iDataplex phase 2 systems.
- Enables rendering in our higher performance nodes that lack GPUs.

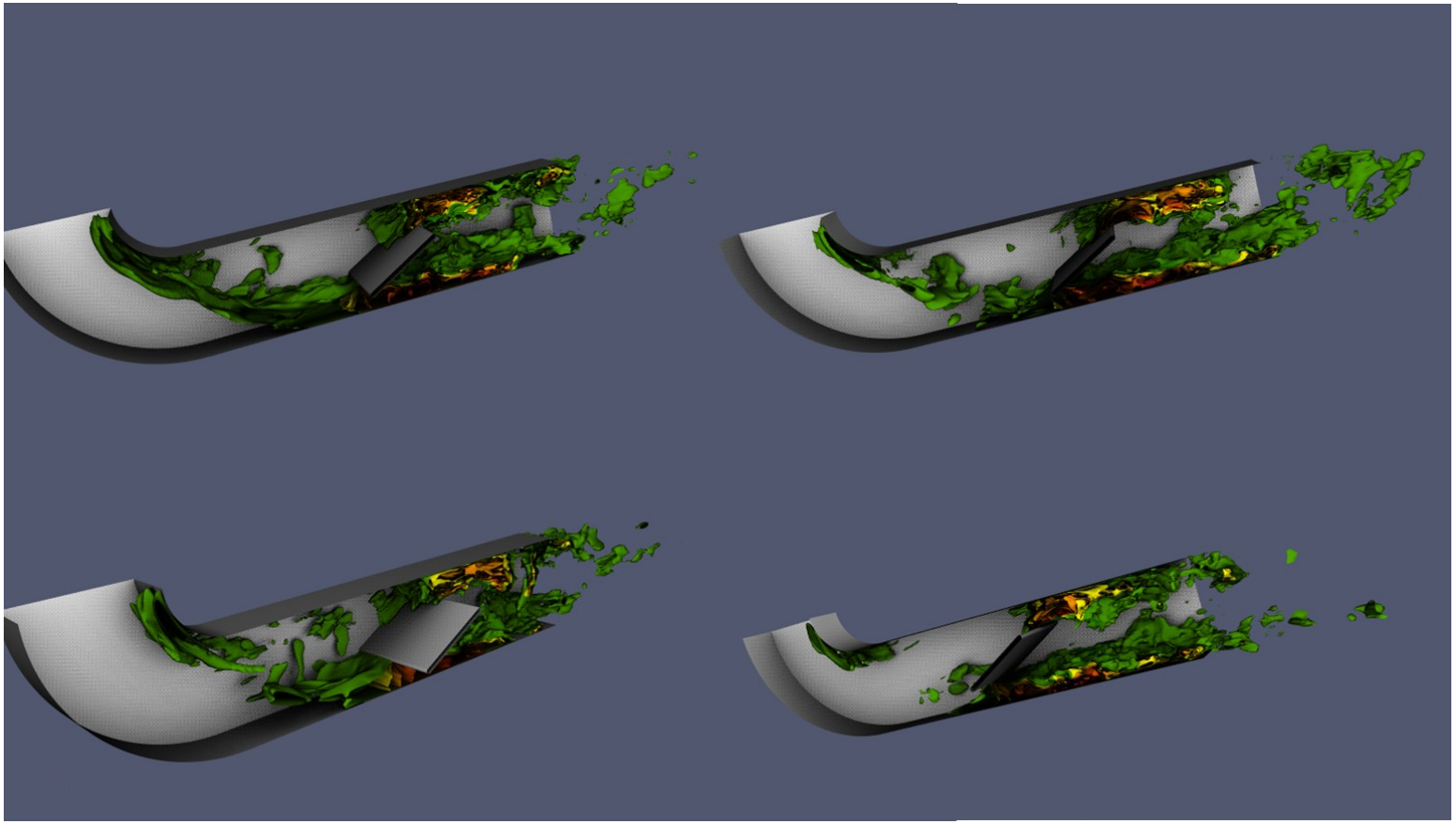




Summary

- KnC development boosts CPU performance.
- Codes need to be highly parallel.
- Optimisations discussed
  - **SIMD performance.**
  - **Efficient arithmetic.**
  - **Threading.**
  - **Library usage.**
  - **Cache tuning.**

Any Questions?



DL_MESO_LBE: Peel/Remainder Loops

- Loads from aligned memory locations produce best performance.
- Unaligned array access results in costly peel loops.
- Peel loop executed until alignment boundary is reached.
- If interation count is not a multiple of vector width scaler remainder loops can be generated.
- Padding arrays to a mulitple of vector width allows fully vectorised code.