Weather and Climate Modeling on GPU and Xeon Phi Accelerated Systems

Mike Ashworth, Rupert Ford, Graham Riley, Stephen Pickles
Scientific Computing Department & STFC Hartree Centre
STFC Daresbury Laboratory

Chris Maynard
HPC Optimisation Team, The Met Office

David Ham
Department of Mathematics, Imperial College

mike.ashworth@stfc.ac.uk
Organisation

HM Government (& HM Treasury)

Department for Business Innovation & Skills (BIS)

RCUK Executive Group

Research Councils UK (RCUK)

Arts & Humanities Research Council (AHRC)

Biotechnology and Biological Sciences Research Council (BBSRC)

Economic and Social Research Council (ESRC)

Medical Research Council (MRC)

Natural Environment Research Council (NERC)

Science & Technology Facilities Council (STFC)
UK Astronomy Technology Centre, Edinburgh, Scotland

Polaris House Swindon, Wiltshire

Chilbolton Observatory Stockbridge, Hampshire

Daresbury Laboratory
Daresbury Science and Innovation Campus
Warrington, Cheshire

Rutherford Appleton Laboratory
Harwell Science and Innovation Campus
Didcot, Oxfordshire

STFC’s Sites

Isaac Newton Group of Telescopes
La Palma

Joint Astronomy Centre Hawaii
• Gung-Ho project
• Unified Model on Intel Xeon Phi
• NEMO on GPUs
• Gung-Ho project

• Unified Model on Intel Xeon Phi

• NEMO on GPUs
GUNG-HO targets a brand new dynamical core

Scalability – choose a globally uniform grid which has no poles (see below)
Speed – maintain performance at high & low resolution and for high & low core counts
Accuracy – need to maintain standing of the model
Space weather implies a 600km deep model
Five year project 2011-2015
Operational weather forecasts around 2020!

From Nigel Wood, Met Office
Design considerations


• Fortran 2003, MPI, OpenMP and OpenAcc
• Other models e.g. PGAS, CAF, are not excluded
• Indirect addressing in the horizontal to support a wide range of possible grids
• Direct addressing in the vertical
• Vertical index innermost is optimal for cache re-use in CPUs, and can also achieve coalesced memory access in GPUs
The Gung-Ho software architecture is structured into layers communicating via a defined API

- **the driver layer** (control for one or more models)
- **the algorithm layer** (high-level specification)
- **the parallelisation system (PSy)** (inter-node and intra-node parallelism, parsing, transformations, hardware-specific code generation)
- **the kernel layer** (toolkit of algorithm building blocks with directives)
- **the infrastructure layer** (generic library to support parallelisation, communications, coupling, I/O etc.)
The arrows represent the APIs connecting the layers.

The direction shows the flow control.
Further development and testing of horizontal [2013]
Testing of proposals for code architecture [2013]
Vertical discretization [2013]
3D prototype development [2014-2015]
Operational around 2020…?
• Gung-Ho project

• Unified Model on Intel Xeon Phi

• NEMO on GPUs
Porting and performance of the UM on Intel Xeon Phi

Preliminary experience and results

Dr Chris Maynard – HPC optimisation team - christopher.maynard@metoffice.gov.uk ext 5038 Floorplate
Contents

• Collaboration between
  • UK Met Office – Chris Maynard
  • STFC Daresbury – Rupert Ford and Mike Ashworth

All results are preliminary
The Unified Model - software

- UM used for both NWP and Climate Models
- Now ~ 25 years old
- Fortran90 (some F77 features remain)
  - Parallelism expressed via MPI
  - Some lower-level OpenMP (retro-fit)
- IO server
  - MPI tasks dedicated to IO
  - Dramatic improvement in IO performance.

UM is ½ million lines of code
ENDGame: scalable dynamical core

At 25km resolution, grid spacing near poles = 75m
At 10km reduces to 12m!

3rd Gen dynamical core (ENDGame) improved scaling – still long-lat
Weak CFL $\Rightarrow \Delta t \downarrow$ as $\Delta x \downarrow$ (implicit scheme)
Data parallel in 2-D
N768 $\Rightarrow$ 1536 x 1152 x 70
Porting the UM

UM has “flat profile”
Lots significant routines → offload mode X
Co-processor only or native mode ✓

Compile with \texttt{--mmic} flag!
Much easier than porting to other accelerator/co-processor

GPU, Cell, FPGA → not really possible
Machines

STFC Daresbury
1 Dual socket 8 core Sandybridge host + KNC
No queues, instant accesses

Stampede, TACC - 6400 nodes!
Queues, batch system, multi-MIC

Met Office IBM Power7 cluster
2 machines
~ 500 32-core nodes each
Unified Model on Intel Xeon Phi ‘Knights Corner’

Tests using radiation code from UM on Intel Xeon Phi (KNC) – pre-production chip

OpenMP not widely used in the UM

Code not optimised for Xeon Phi

Speedup of the UM N48 Atmosphere model on 2*Sandybridge, Xeon and KnC

- 1xn MPI 1 OpenMP 2*Sandybridge
- 1 MPI n OpenMP 2*SandyBridge
- linear Sandybridge
- 1xn MPI 1 OpenMP Xeon E31270 (gfortran)
- mnx MPI 1 OpenMP KnC
- linear KnC (16)

Total parallelism (mpi tasks * openmp threads)
96x72x70 single Phi – UM

6x10 MPI tasks,
KMP_AFFINITY="verbose,granularity=thread,compact"

A: 1 OMP thread
B: 2 OMP, -O2, -O3 and compile thread=2
C: 2 OMP, -mkl, seg-size 20-120
D: 4 OMP, seg-size
E 2x Phi 6x10 each – 2 OMP
Interpreting the results

- Threading performance is poor
  - Segments for radiation routines doesn’t help
  - Is there enough work per thread?
- MKL really helps radiation routines
  - Lots of trig func - library boosts performance 30%
- Use of two KNC cards is faster
  - Even for MPI over PCIe - enough data parallelism
  - OMP at low level in UM – incomplete coverage
- Bigger problem size?
  - 192x144 runs out of global memory on 1 and 2 cards
  - Same local volume size as N48 on 4!
Focus on the solver - threading

Solver performance is dominated by preconditioner – **tri_sor**

Red-black checkerboard for threading

N48 – 96 x 72 / 6 x 10 (MPI) = **16 x 7(8)** per task

split over threads?

Serial stand-alone solver – local volume **100x100**

c.f. with single socket (8-core) sandy bridge

Same global volume \( \rightarrow 250x300 \) local volume
Threaded works very well
Good thread performance from KNC
Faster than SandyBridge 1 hyperThread (HT)
Is HT enabled? – 2HT same as 1HT
Much better than SMT on Power7

KNC 100x100 per core x 60 cores = 600000
SB/PW7 250x300 per core x 8 cores = 600000

SB/Pwr7 faster than KNC on smaller volumes
Focus on the solver - Vectorisation

SIMD unit is 512 bytes wide, 32 vector registers
Data alignment, prefetch, compiler directives - compiler vectorise loop
-vec-report compiler flag
   Critical loops in tri_sor are vectorising
   But checkboarding \(\rightarrow\) stride 2 access
-S compiler flag – peer at assembler code
   tri_sor.F90 188 lines tri_sor.s is 3248 lines!

High register numbers are not heavily occupied
30 vector FMAs, 14 vector multiplies, 10 vector adds
Vector unit is used, but not fully stressed by code
Conclusions

Porting UM to Intel Phi in native mode is easy
Initial performance is disappointing
MKL library boosts performance of certain
routines

**Solver**
Thread performance is good for ↑local volumes (LV)
Small global memory enforces LV↓ – performance ↓ Compiler is vectorising thread friendly data layout compromises vector performance

Solver is only ~¼ of timestep: slow physics, solver, fast physics, advection

Data layout is critical “Optimizing Lattice QCD on Intel(R) Xeon Phi(tm) Coprocessor” B.Joo ISC13

Cost of data re-ordering is high for legacy code – get it right for Gung-ho!
Single precision solver doubles performance – reduces data reorder cost
Might be worth cost – watch this space
Outline

• Gung-Ho project

• Unified Model on Intel Xeon Phi

• NEMO on GPUs
NEMO is the leading ocean model in use in Europe with a multi-national development team

GNEMO is a 12-month project to assess feasibility of porting to GPUs: Jan-Dec 2011

Ported two routines, using 3 programming models:
- lateral diffusion of tracers (*traldf_iso*); modest speed-up: 6-core Westmere CPU = 69% of Tesla GPU
- ice rheology (*lim_rhg*); slower on GPU

**Issues:** limited parallelism, low computational intensity, frequent halo exchanges
Flat profile (hottest subroutine 6%)

code converted using OpenACC directives

41 files are modified

63 subroutines called from step.F90 are processed

142 arrays are kept on the GPU for the duration of the run

Changes to source file include:

• MPI communication
• Rearranging loops to make them parallelizable
• Introducing temporary arrays for sub-arrays and expressions
• Writing output files is switched off for the time being
NEMO Acceleration on GPUs using OpenACC

Maxim Milakov, Peter Messmer, Thomas Bradley, NVIDIA

GYRE only – we are looking at more realistic test cases, with ice and land

Tesla M2090 GPUs 6GB RAM, Westmere CPUs 2.93 GHz

Milakov, Messmer, Bradley, GTC, 18-22 March 2013
• There are discussions about using the Gung-Ho computational framework
  • development of a new ocean model or
  • a new version of an existing ocean model
• Case based on leveraging investment in Gung-Ho (£7.5M = $12M)
• Gung-Ho presented to the NEMO Enlarged Developers' Committee, 18th June 2013, Paris
• Proposal submitted on Monday 16th September for a technology proof-of-concept project
We acknowledge use of **Hartree Centre** resources in this work. The STFC Hartree Centre is a research collaboratory in association with IBM providing High Performance Computing platforms funded by the UK's investment in e-Infrastructure. The Centre aims to develop and demonstrate next generation software, optimised to take advantage of the move towards exascale computing.

**Gung-Ho** work was supported by the Natural Environment Research Council Next Generation Weather and Climate Prediction Programme (grant numbers NE/I022221/1 and NE/I021098/1) and the Met Office

**GPU Acceleration of the NEMO Ocean Model (GNEMO)** work was supported by the Natural Environment Research Council through the Technology Proof-of-Concept grant number NE/I00095X/1
Thank you for listening

Mike Ashworth
mike.ashworth@stfc.ac.uk
http://www.stfc.ac.uk/scd