Exploiting Hybrid Parallelism in LBM Implementation Musubi on Hawk

WSSP 2021-03-17
Harald Klimach, Kannan Masilamani, Sabine Roller

harald.klimach@uni-siegen.de
Hybrid Parallelisation

• Increasing large numbers of cores per node:
  – Hawk has 128 cores per node!

• MPI only results in large number of individual connections from single node to others

• With OpenMP we get: larger partitions, less fragmented communication
OpenMP Overheads

- Unfortunately the use of threads for OpenMP parallelism introduces some overheads
  - Run into the risk of cache thrashing or similar memory effects and actual slow down
- MPI communication itself usually does not benefit from OpenMP parallelism
Lattice Boltzmann Method

• Fluid simulations with the idea of using cellular automata for the discretization

• Utilizes discrete velocity vectors with particle distribution functions

• Allows for fairly simple representation of obstacles
Lattice Boltzmann Method

• Fluid simulations with the idea of using cellular automata for the discretization

• Utilizes discrete velocity vectors with particle distribution functions

• Allows for fairly simple representation of obstacles
Lattice Boltzmann Method contd.

• „Particles“ are streamed to neighbors and a collision operation determines new distribution function

• Results in classical stencil code

• Enables explicit in time evolution of incompressible flow equation (small computational effort per timestep)
Musubi

- Open Source Lattice Boltzmann solver
- Local refinement (multi-level)
- Multiple Species (Maxwell-Stefan)
- Turbulence modelling

https://apes.osdn.io/pages/musubi.html
Mesh Representation with Treelm

![Diagram showing mesh representation and space-filling curve with nodes and levels labeled from 0 to 5. The treelIDlist: Representation on Disk is also shown with numbers and colors indicating node connections.](image-url)
Musubi Properties

• Mostly written in Fortran 2003

• Primarily parallelized with MPI

• Double buffer for the state, additional arrays for auxiliary quantities

• Computational intensity around 3 Bytes/FLOP
OpenMP in Musubi

• Still in the process of being rolled out

• Advantage of OpenMP: can be partially deployed for important parts and then expanded to include ever more parts of the code

• Basic kernels straight forward to parallelize with OpenMPs parallel do
Space-Filling Curve and OpenMP

- Mesh elements are processed in the loop according to the ordering of the space-filling curve
- Splitting loops into blocks for OpenMP threads results in partitions similar to the MPI partitioning of the mesh
  - Conflicting memory access is avoided naturally
Disadvantages

• Some parts not OpenMP parallel yet, users may not be aware of this

• Some parts hard to parallelize with OpenMP

• OpenMP loop parallelization competes with vectorization

• Some operations do not benefit from OpenMP, but would benefit from more MPI processes
Hawk Node Info

- 2x AMD EPYC 7742 64-Core Processor
- 2247 MHz
- 128 physical cores

- 8 NUMA nodes (16 cores each)
  
  Considering up to 16 OpenMP Threads (in 8 MPI processes per node)
Performance Measure for LBM

- It is common in LBM to consider performance in terms of million lattice updates per second (MLUPS)

- Provides an intuitive number to assess performance akin to MFLOPS
BGK D3Q19 Single Node (128 cores) on HAWK

Always Utilizing all 128 cores (number of MPI processes accordingly)
BGK D3Q19 8 Nodes on HAWK

![Graph showing MLUPS per node vs. Elements per node for different thread counts (1, 2, 4, 8, and 16 threads).]
BGK D3Q19 64 Nodes on HAWK

Simulation Techniques & Scientific Computing

OpenMP Parallelism in Musubi
BGK D3Q19 Weak Scaling 16M elements $pN$
BGK D3Q19 4 Threads per Process on HAWK

- **MLUPS per node**
- **Elements per node**
- **Graph**
  - 1 Node
  - 8 Nodes
  - 64 Nodes

Simulation Techniques & Scientific Computing
D3Q27 Stencil

- Requires exchange with all 26 neighbors
  - Larger communication surface

- Needed for some collision kernels
BGK D3Q27 Single Node on Hawk

27 directions require more memory -> less elements per node
BGK D3Q27 8 Nodes on Hawk

MLUPS per node

Elements per node

1 Thread
2 Threads
4 Threads
8 Threads
16 Threads
BGK D3Q27 64 Nodes on Hawk

![Graph showing MLUPS per node vs. Elements per node for different thread counts. The graph illustrates the performance of BGK D3Q27 on 64 nodes using 1, 2, 4, 8, and 16 threads. The x-axis represents the number of elements per node, ranging from $10^3$ to $10^7$, and the y-axis represents MLUPS per node, ranging from 0 to 350.]
BGK D3Q27 Weak Scaling 16M Elements pN

![Graph showing MLUPS per node vs. Number of nodes for different thread counts. The graph plots 8^0, 8^1, 8^2 nodes with line markers for 1, 2, 4, 8, and 16 threads. The data points indicate a trend towards stable performance as the number of nodes increases.](image)
BGK D3Q27 Strong Scaling

MLUPS per node

Number of nodes

1 Thread
2 Threads
4 Threads
8 Threads
16 Threads
Conclusion

• Up to 8 threads provide reasonable performance on Hawk with the current state of OpenMP parallelism in Musubi

• Larger number of threads not beneficial at the moment

• OpenMP parallelism important for larger stencil (D3Q27)

• Strong Scaling improved by hybrid parallelism
Thank you!