Alleviating memory bandwidth pressure with wavefront temporal blocking and diamond tiling

Tareq Malas*§
Georg Hager§
Gerhard Wellein§
David Keyes*

§Erlangen Regional Computing Center, Germany
*King Abdullah Univ. of Sci. & Tech. (KAUST), Saudi Arabia
Memory bandwidth starved stencil computations, with spatial blocking

- Example system
  - Double precision
  - Domain size: $520^3$
  - 10-cores Intel Ivy Bridge
  - 7-pt 3D stencil with spatial blocking
  - Constant symmetric coefficients
- Performance limit =
  \[
  \frac{\text{attainable memory BW}}{\text{Memory Bytes / LUP}} = \frac{40 \text{ GB} / \text{s}}{24 \text{ B} / \text{LUP}} = 1.67 \text{ GLUP} / \text{s}
  \]
Stencil computations challenges in future architectures

• Each node may have up to a thousand shared-memory cores with:
  – small cache size per core
  – small memory bandwidth per core
  – complex cache sharing among cores
  – expensive synchronization among all the cores
  – interaction between heterogeneous processors

• Expensive synchronization after each iteration
Diamond tiling

- Improves the performance on shared and distributed memory systems

- High data reuse, reducing memory accesses
- Provides independent space-time blocks to reduce synchronization between threads and nodes
- Overlap computation with communication
- Tessellation reduces the overhead of handling the boundaries of subdomains, sockets, and heterogeneous processors
Diamond tiling

- Improves the performance on shared and distributed memory systems
  - High temporal reuse, reducing memory accesses

Diagram showing the concept of diamond tiling with time and space axes.
Diamond tiling

• Improves the performance on shared and distributed memory systems
  – high temporal reuse, reducing memory accesses
  – provides independent space-time blocks to
    • reduce synchronization between threads and nodes
    • overlap computation with communication
Diamond tiling

- Improves the performance on shared and distributed memory systems
  - high temporal reuse, reducing memory accesses
  - provides independent space-time blocks to
    - reduce synchronization between threads and nodes
    - overlap computation with communication
  - tessellation reduces the overhead of handling the boundaries of subdomains, sockets, and heterogeneous processors
Wavy assignments for larger diamonds

Send Left

Send Right

O : Out data
I : In data
W: wait
Related work

• Diamond tiling technique has drawn the attention of the research community in recent years
  
  
  
  
  
  
1-Core-Wavefront temporal blocking + Diamond tiling (1CWD)

- Extruded diamonds
  - Diamond tiling and domain decomposition across the Y-axis
  - Wavefront temporal blocking along the Z-axis
  - No decomposition across the X-axis

Multi-Core-Wavefront temporal blocking + Diamond tiling (MCWD)

- Advantages:
  - Can run at small domain sizes on many-core architectures, as it does not require concurrent tile for each thread
  - Large reduction in cache size requirements compared to having 1 cache block per core
  - Utilizes the shared cache between cores and hardware threads of modern processors
Diamond tiling temporal blocking

• Setup
  – Double precision
  – Domain size: $520^3$
  – 10-cores Intel Ivy Bridge

• MCWD achieves
  – 1.11x speedup over 1CWD
  – 2.14x speedup over spatially blocked code
Coming enhancement

• Assign threads to multiple groups instead of one large threads group
• Use hierarchical cache blocking to improve the data reuse at different cache levels