Implementing BCs in Legion-S3D

Hemanth Kolla
Sandia National Laboratories

Legion Bootcamp
December 7th, 2015
Stanford, CA
Background

- S3D is an explicit finite difference PDE solver for turbulent combustion:
  - Rectangular Cartesian fixed structured mesh.
  - Perfectly load balanced spmd decomposition.
  - Very large field space (~100 state variables at each grid point).
  - Number of PDEs (number of stencil ops) scales with #fields.
  - Rich mix of physics kernels involved in the r.h.s function evaluation.
Background

- All of S3D r.h.s.f was ported to Legion (Mike/Sean/??).
- Interoperate with base S3D (Fortran-MPI).
Background

- All of S3D r.h.s.f was ported to Legion (Mike/Sean/??).
- Interoperate with base S3D (Fortran-MPI).
- “Manual spmd”-fication. Task hierarchy:
  - Top level task (single)
Background

- All of S3D r.h.s.f was ported to Legion (Mike/Sean/??).
- Interoperate with base S3D (Fortran-MPI).
- “Manual spmd”-fication. Task hierarchy:
  - Top level task (single)
    - “distribute task” (one per spmd sub-domain) with phase_barriers.
Background

- All of S3D r.h.s.f was ported to Legion (Mike/Sean/??).
- Interoperate with base S3D (Fortran-MPI).
- “Manual spmd”-ification. Task hierarchy:
  - Top level task (single)
    - “distribute task” (one per spmd sub-domain) with phase_barriers.
    - All physics tasks (stencils, chemistry, analytics...)

Can you summarize the background for the project?
Background

- All of S3D r.h.s.f was ported to Legion (Mike/Sean/??).
- Interoperate with base S3D (Fortran-MPI).

“Manual spmd”-fication. Task hierarchy:
- Top level task (single)
  - “distribute task” (one per spmd sub-domain) with phase_barriers.
    - All physics tasks (stencils, chemistry, analytics...)

- Single compiler for chemistry and transport kernels.
- The legion port operational for periodic problems.
Scenario for physical boundaries

- Boundary type 1
- Boundary type 2

interior
• 7+ different types of BCs in S3D.
• 7+ different types of BCs in S3D.
• All have similar region requirements, but differ in the math.
• 7+ different types of BCs in S3D.
• All have similar region requirements, but differ in the math.

Option A

• Implement/register tasks for each BC type (a.k.a MiniAero).
• Have the right one instantiated at runtime based on problem type.
- 7+ different types of BCs in S3D.
- All have similar region requirements, but differ in the math.

<table>
<thead>
<tr>
<th>Option A</th>
<th>Option B</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Implement/register tasks for each BC type (a.k.a MiniAero).</td>
<td>• Have a generic BC task.</td>
</tr>
<tr>
<td>• Have the right one instantiated at runtime based on problem type.</td>
<td>• Branch statements for implementing the BC type required by the problem.</td>
</tr>
</tbody>
</table>
• Color and partition each spmd subdomain into int/bt1/bt2/.../bt7.
• Color and partition each spmd subdomain into int/bt1/bt2/.../bt7.
• Launch all bc_type tasks for all spmd subdomains.
• Color and partition each spmd subdomain into int/bt1/bt2/.../bt7.
• Launch all bc_type tasks for all spmd subdomains.
• BC types that do not exist => NO_REGION requirement => no task instance.
Observations

• The runtime will instantiate a task even if all its region requirements are NO_REGIONs.
Observations

• The runtime will instantiate a task even if all its region requirements are NO_REGIONs.

• Task might need to operate on arguments it receives.
Observations

- The runtime will instantiate a task even if all its region requirements are NO_REGIONs.

- Task might need to operate on arguments it receives.

- What is the nature of an index space task that has only arguments but NO_REGION requirements?
Observations

• The runtime will instantiate a task even if all its region requirements are NO_REGIONNs.

• Task might need to operate on arguments it receives.

• What is the nature of an index space task that has only arguments but NO_REGION requirements?

• What is the nature of an index space task whose region requirements are all READ_ONLY?
Iterating over an index_space or Rect based off of a NO_REGION causes runtime error. Users need to check for this.

Projection functions could be better documented, maybe with a good example.

In S3D, BCs require specialized stencils (in_plane, one-sided, variable-width) which is hard/tricky with explicit ghost zones.

Peeking into the runtime (legion.*, runtime.*) is not hairy and occasionally very helpful.
Thank you