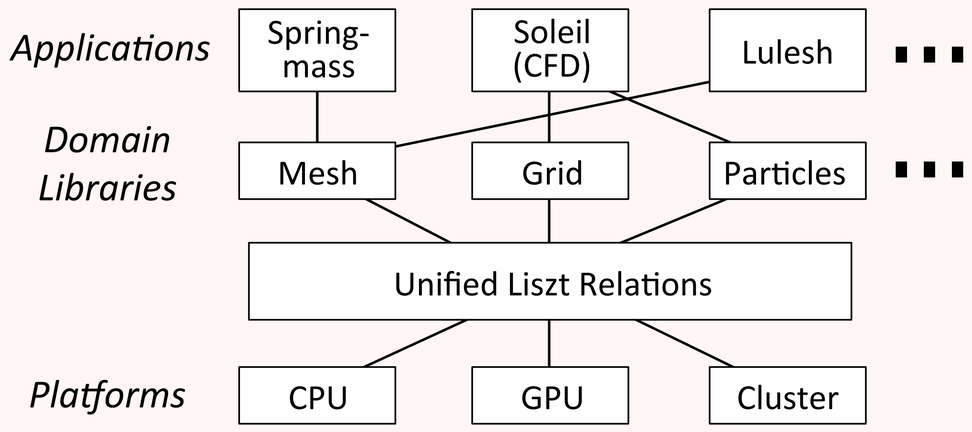
Liszt Updates

Chinmayee Shah, Gilbert Bernstein, Elliott Slaughter, Sean Treichler
Philip Levis, Alex Aiken, Pat Hanrahan

PSAAP II
Particle-laden Turbulence in a Radiation Environment

Language Design



Example Code

```
local Tetmesh = require 'domains.tetmesh'
local dragon  = Tetmesh.Load 'dragon.veg'
```

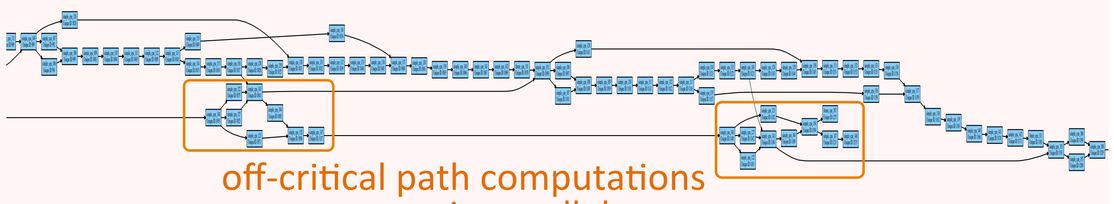
```
dragon.edges:NewField('stiffness',L.float):Load(2)
dragon.vertices:NewField('acc',L.vec3f)    :Load({..})
...
```
define fields of data

```
local liszt ComputeForces(v : dragon.vertices)
  var force : L.vec3f = {0,0,0}
  for e in v.edges do
    var diff = e.head.pos - v.pos
    force -= e.stiffness * (e.rest_len - diff)
  end
  v.acc = force / v.mass
end
...
```
define per element computation

```
for i = 1, 300 do
  dragon.vertices:foreach(ComputeForces)
  ...
end
```
sequence of data parallel computations

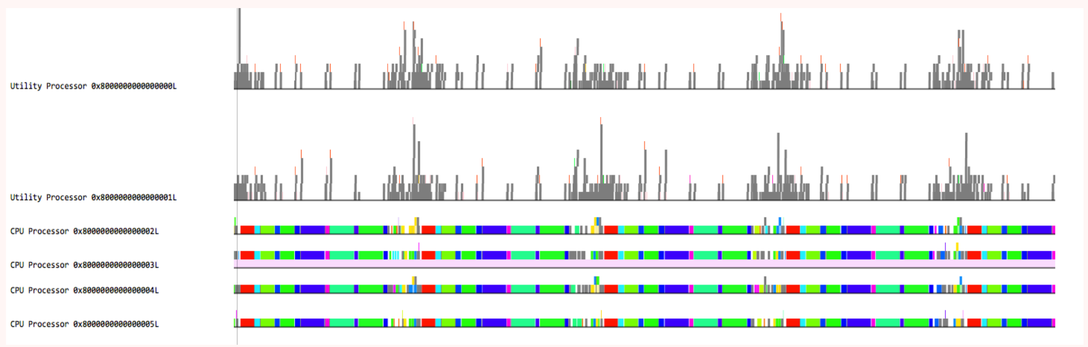
Legion Integration

Liszt relations → Legion logical regions
Liszt computations → Legion tasks



off-critical path computations can run in parallel

Legion Multi-Core Results



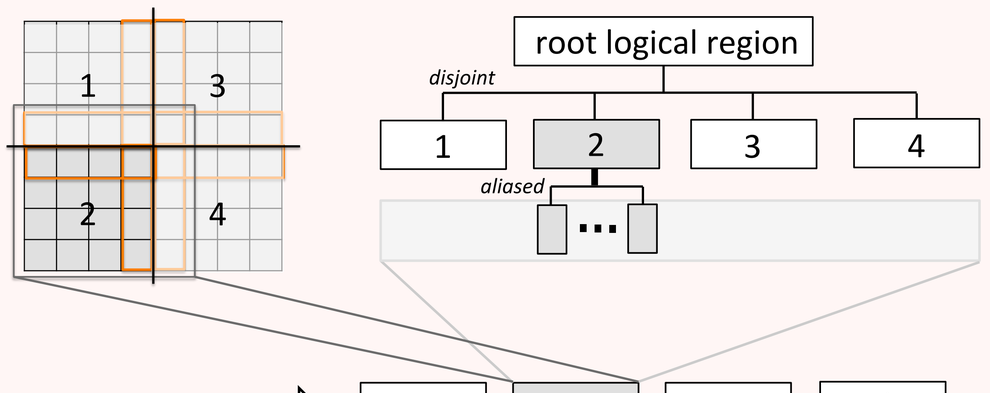
1 partition	19.7s/step
2 partitions	11.2s/step (1.8x)
4 partitions	5.7s/step (3.5x)

Partitioning Across Nodes

Computation: `c(1,0).velocity – c(-1,0).velocity`

↓ Stencil Analysis

Ghost Regions → Partition Tree



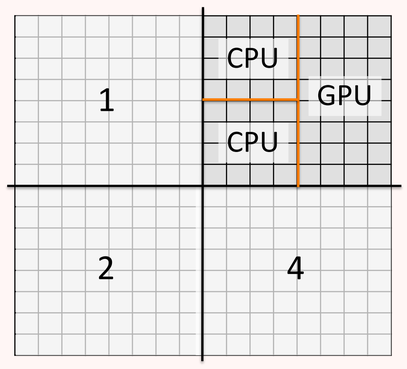
`foreach` call → Task 1 | Task 2 | Task 3 | Task 4

Liszt Computations → computations over ghost + central regions

Planner

(typed) computation + relation → partition data

How to partition across & within nodes?



What ghost regions to use?

- Determine ghost strategy after aggregating different access patterns
- Multiple instances vs one common instance on local node (shared memory)

Legion Multi-Node Results