An Active Library for Semi-Regular Grids

Andrew Stone, Michelle Mills Strout

Colorado State University

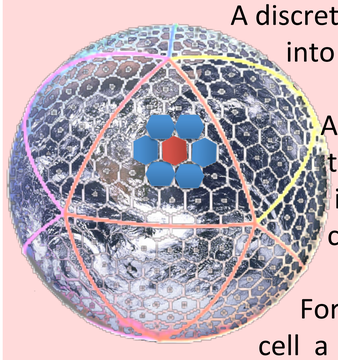
FRCRC
HPC 2012

Motivation

Scientists use Earth Simulation programs to:
- **Predict** change in the weather and climate
- **Gain insight** into how weather and climate works

These programs contain components for:
ocean, land-surface, sea ice, atmosphere

These programs work on a **discretization** of the Earth (called a **Grid**) and apply **stencil computations**.



A discretization partitions the earth into a finite number of cells

A stencil computation iterates through each cell and updates it using values from surrounding cells.

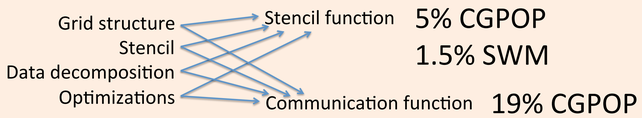
For example: to update the red cell a combination of values from the blue cells is used

Code for grid structure impacts the stencil code. The code is further complicated when it to be run on a parallel machine and optimized.

Tangled Details

```
Loop {
   communicate()
   stencil()
}
```

Stencil and communication code is commonly impacted by grid structure and parallelization details.

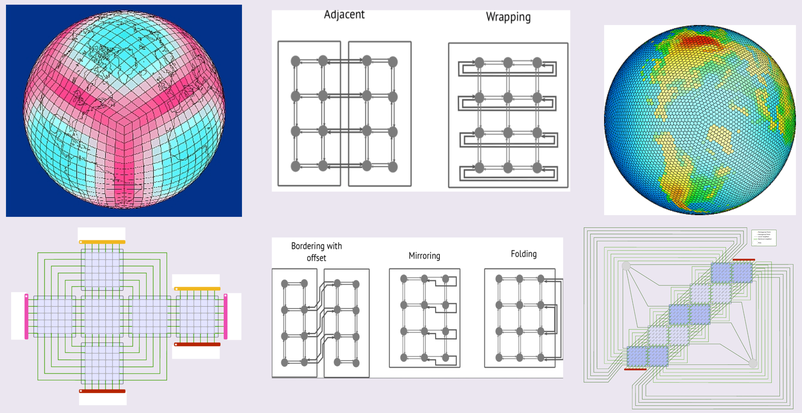


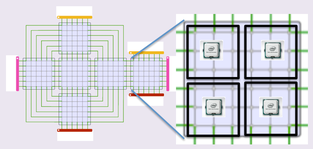
5% CGPOP
1.5% SWM

19% CGPOP
30% SWM

Earth Grids

- Many Earth Simulation programs use grids that consist of **sets of regular grids connected in an irregular fashion**
- We call this class **Semi-Regular**
- Below we illustrate the cubed sphere and geodesic grid, and various connectivity patterns:



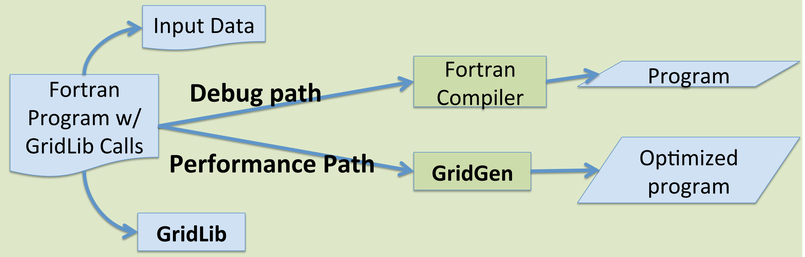


After data is structured into a grid it is blocked and distributed to processors.

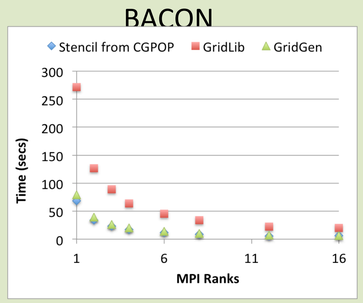
Example Program

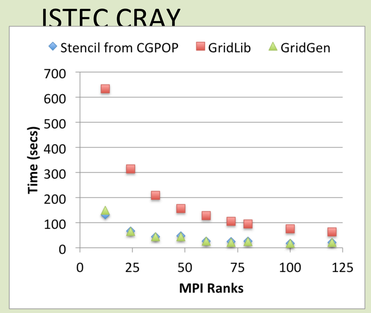
```
! Define grid
call sugrid_new(sg1, N, M)
call sugrid_new(sg2, N, M)
call grid_new(g)
call grid_placeAdjacentWE(g, sg1, sg2)
! Define distribution and read in data
call distribution_new_blocked(dist, g, 100, 100)
call data_new_from_file(data_in, “input.dat”, g, dist)
! Apply stencil
data_out = data_apply(data_in, fourPtAvg)

real function fourPtAvg(A, i, j)
    fourPtAv = &
        0.2 * (A(i,j) + A(i-1,j) +  &
        A(i+1, j) + A(i, j-1) +  &
        A(i, j+1))
end function
```

GridGen



With the GridGen code generator we are able to produce semi-regular grid stencil code that matches hand-written code.





- Experiments are with CGPOP stencil on a Dipole grid
- Bacon is: a 16 core, 2.13 GHz, XeonE7 machine
- Bacon experiments ran for 100 iterations on a 3600 x 3600 grid
- The ISTEC Cray is: A Cray XT6m with 1248 cores across 52 nodes, 24 cores per node
- ISTEC Cray results ran for 1000 iterations on 10800 x 10800 grid

Future Work

- Non compact stencils
- Implementation in SWM and CGPOP
- Optimizations