Separating Implementation Concerns in Semiregular Grids for Stencil Computations

Andy Stone

July 2nd, 2013

PhD Defense

Colorado State University

Introduction

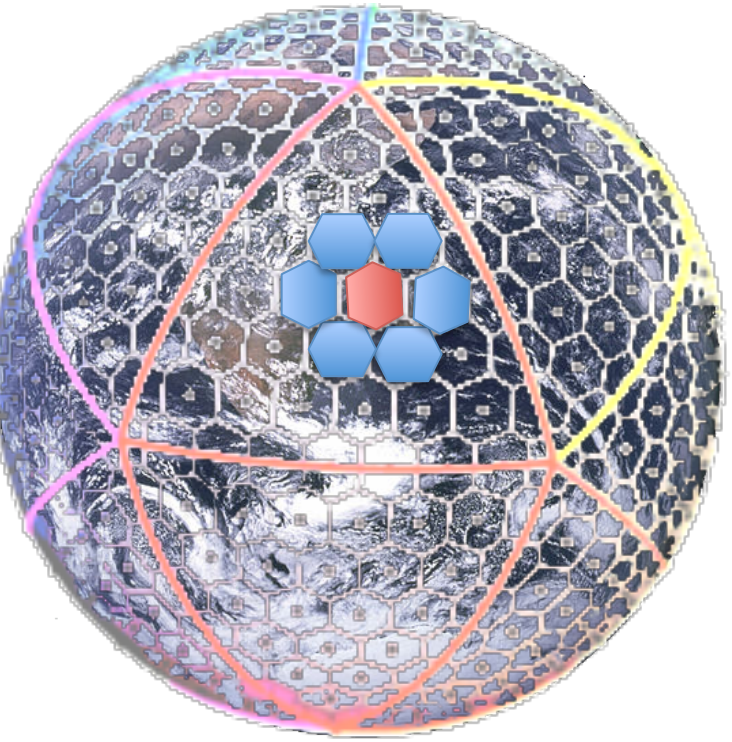
- **Goal:** to improve programmability and maintainability of Earth simulation applications.
- **Importance:** scientists use these applications to:
  - **Predict** change in the weather and climate.
  - **Gain insight** into how the climate works.
- The more time spent developing and maintaining an application, the less time remains to actually use it.
- **Approach:** Develop a new programming model and library called GridWeaver.
- **Case studies:** Modify the code of two proxy applications to use GridWeaver.
  - Proxy applications are smaller models of important portions of larger applications.
  - **CGPOP**: Proxy of POP (an ocean simulation code).
  - **SWM**: Proxy of GCRM (a cloud resolution model).

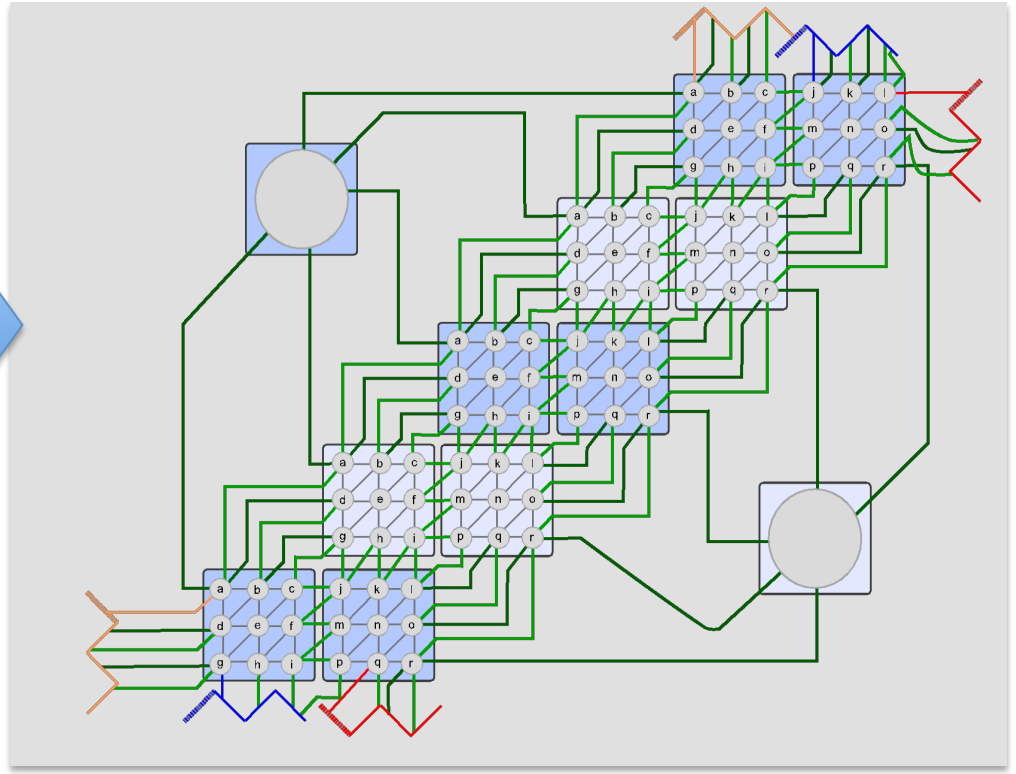
Background:

How do Earth simulation applications work and what affects their programmability and modifiability?

From a computer science perspective: what do Earth Simulation applications do?

They solve PDE's on semiregular grids with stencil computations.





Grid connectivity affects programmability

```
DO j = 2,jm-1
  DO i = 2,im-1
```

Iterate through nodes in subdomain

Apply stencil

```
    x2(i,j) = area_inv(i,j) *
      ((x1(i+1,j)-x1(i,j))*laplacian_wghts(1,i,j.nsd) +
       (x1(i+1,j+1)-x1(i,j))*laplacian_wghts(2,i,j,nsd) +
                                    . . .
  ENDDO
ENDDO
```

Update north pole

```
! NORTH POLE
i = 2; j = jm
x2(i,j) = area_inv(i,j)*laplacian_wghts(1,i,j)*((x1(i+1,j)- . . .
```

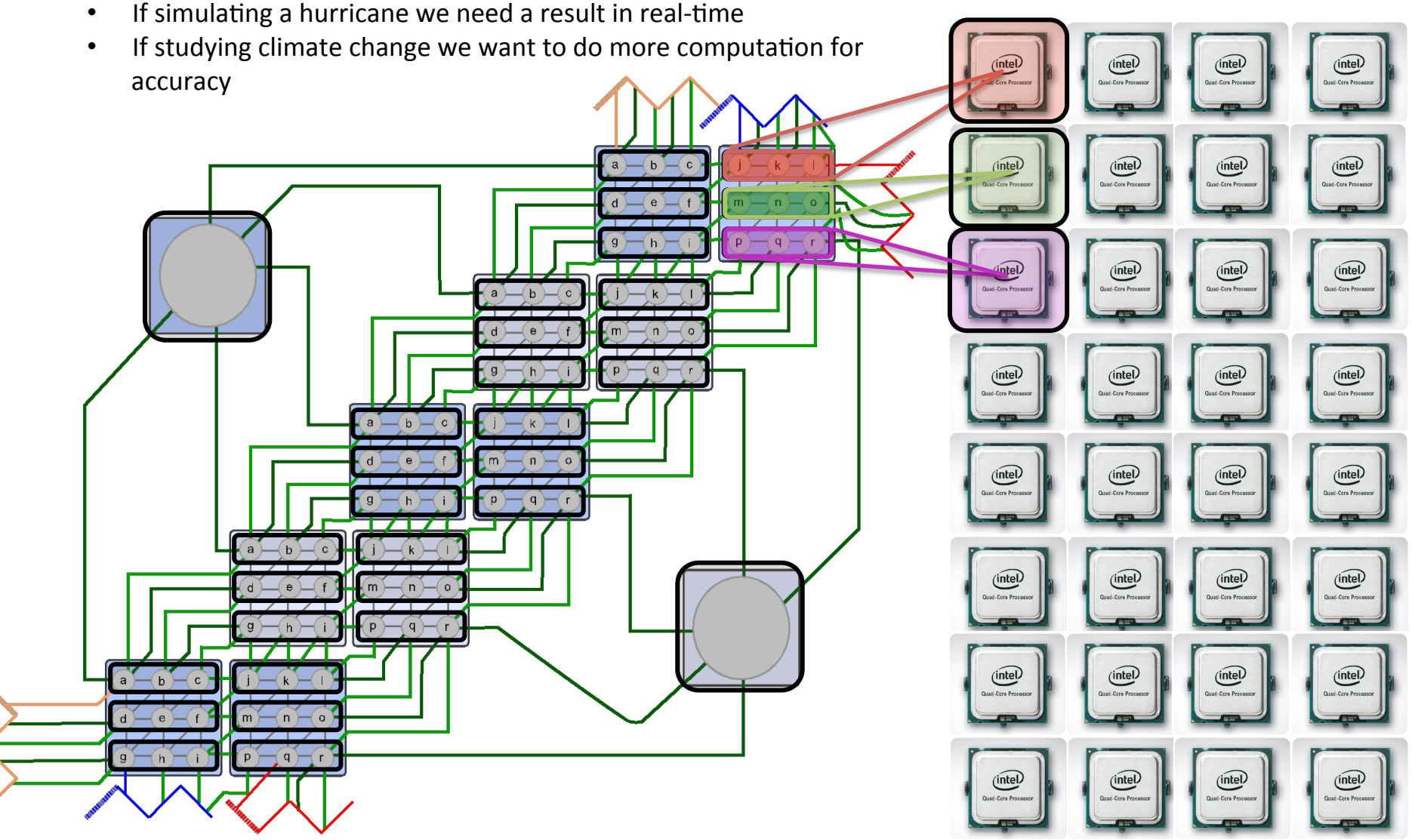
Update south pole

```
! SOUTH POLE
i = im; j = 2
x2(i,j) = area_inv(i,j)*laplacian_wghts(1,i,j)*((x1(  1, jm)- . . .
```

Parallelism affects programmability

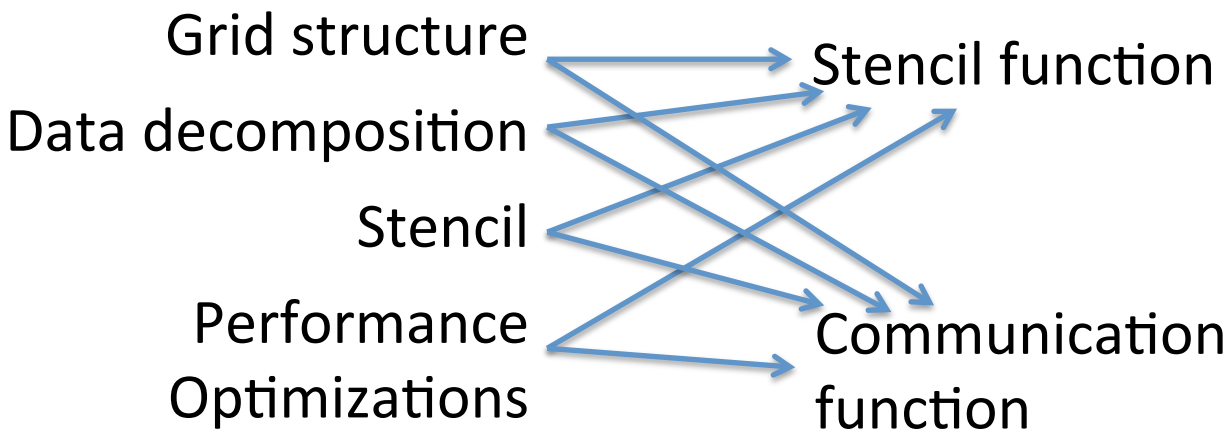
Performance is crucial:

- If simulating a hurricane we need a result in real-time
- If studying climate change we want to do more computation for accuracy



Tangled Specifications affect programmability

```
initialization()

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

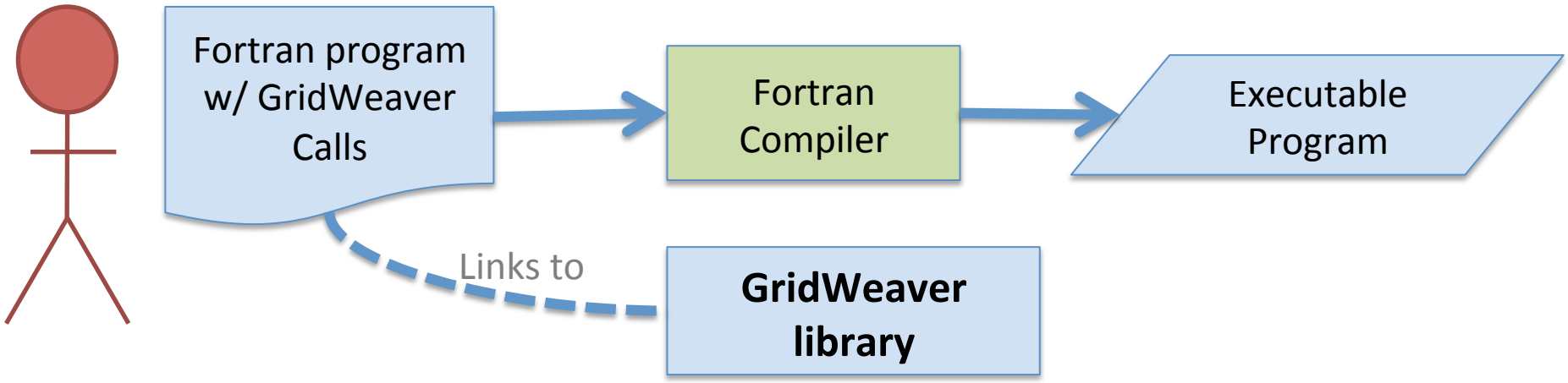


5% (150 SLOC) **CGPOP**
12% (1770 SLOC) **SWM**

38% (1226 SLOC) **CGPOP**
6.8% (1010 SLOC) **SWM**

Are modifications actually made: POP grid changed between versions 1.4.3 and 2.0.

Approach for improving programmability



- Since GridWeaver is a library users can integrate it into existing programs or write new programs from scratch.

Approach for improving programmability

Punchline: Separates concerns and automates communication

Structure of a GridWeaver program:

```
- Grid connectivity spec
- Decomposition spec
- Initialize data

loop {
  communicate()
  stencil()
}
```

- Communication function provided by GridWeaver.
- Instead of writing 1226 lines of code for CGPOP and 1010 lines of code for SWM, with GridWeaver users write **0 lines of code.**

Simple stencil function:

```
STENCIL(swmStencil, x2, x1, area_inv, laplacian_wghts, i, j)
   x2(i,j) = area_inv(i,j) *
     ((x1(i+1,j)   - x1(i,j)) * laplacian_wghts(1,i,j) +
      (x1(i+1,j+1) - x1(i,j)) * laplacian_wghts(2,i,j) +

end function
```

Thesis Contributions

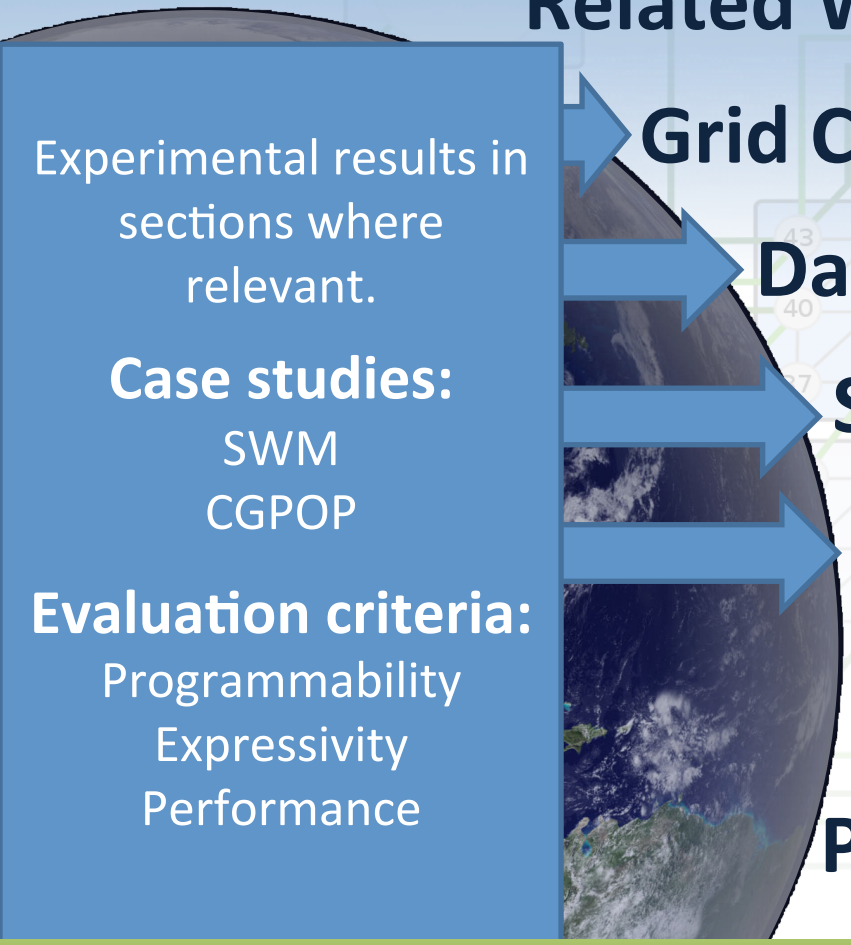
In today's talk: New programming model for semiregular grids

- GridWeaver active library (library + source-to-source translator tool).
- Abstractions: grid connectivity, decomposition, stencil computation.
- Communication plan generation algorithms.
- Case studies that evaluate expressibility, performance, and programmability of GridWeaver.

In thesis: Methodology for evaluating programming models

- Terminology for comparing programming models.
- Development of CGPOP as proxy application of POP.

Outline



Related Work

Grid Connectivity

Data\Computation Distribution

Stencil Computations

Communication

Conclusions

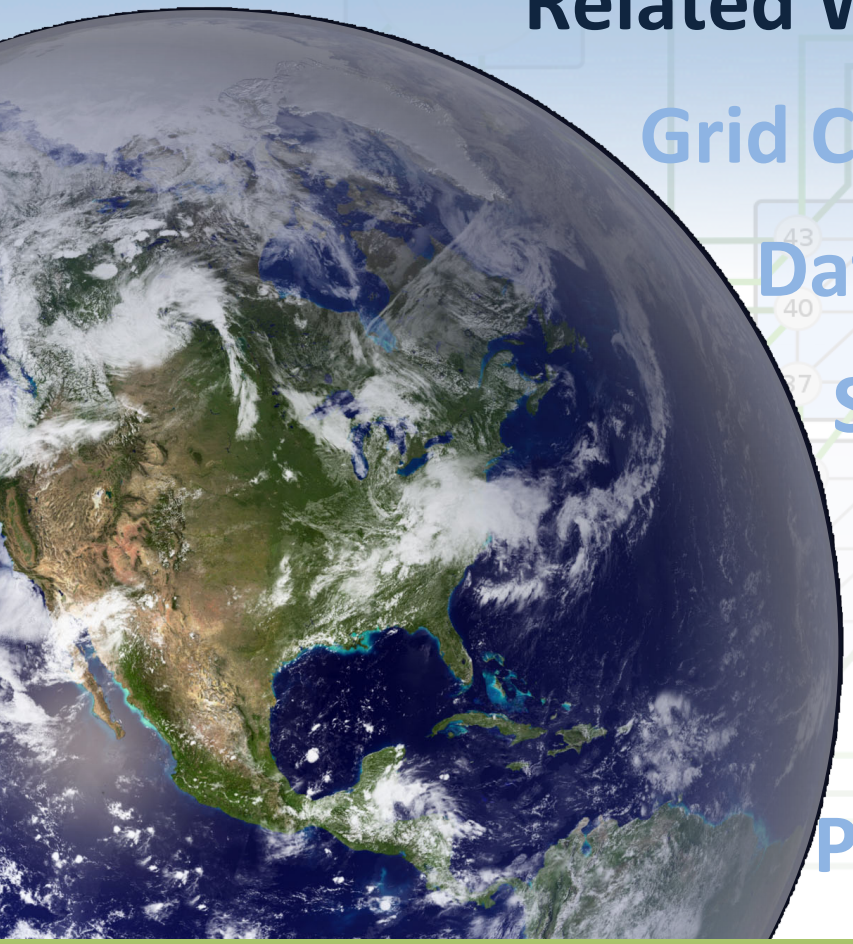
Publication Record

Experimental results in sections where relevant.

Case studies:
SWM
CGPOP

Evaluation criteria:
Programmability
Expressivity
Performance

Outline



Related Work

Grid Connectivity

Data\Computation Distribution

Stencil Computations

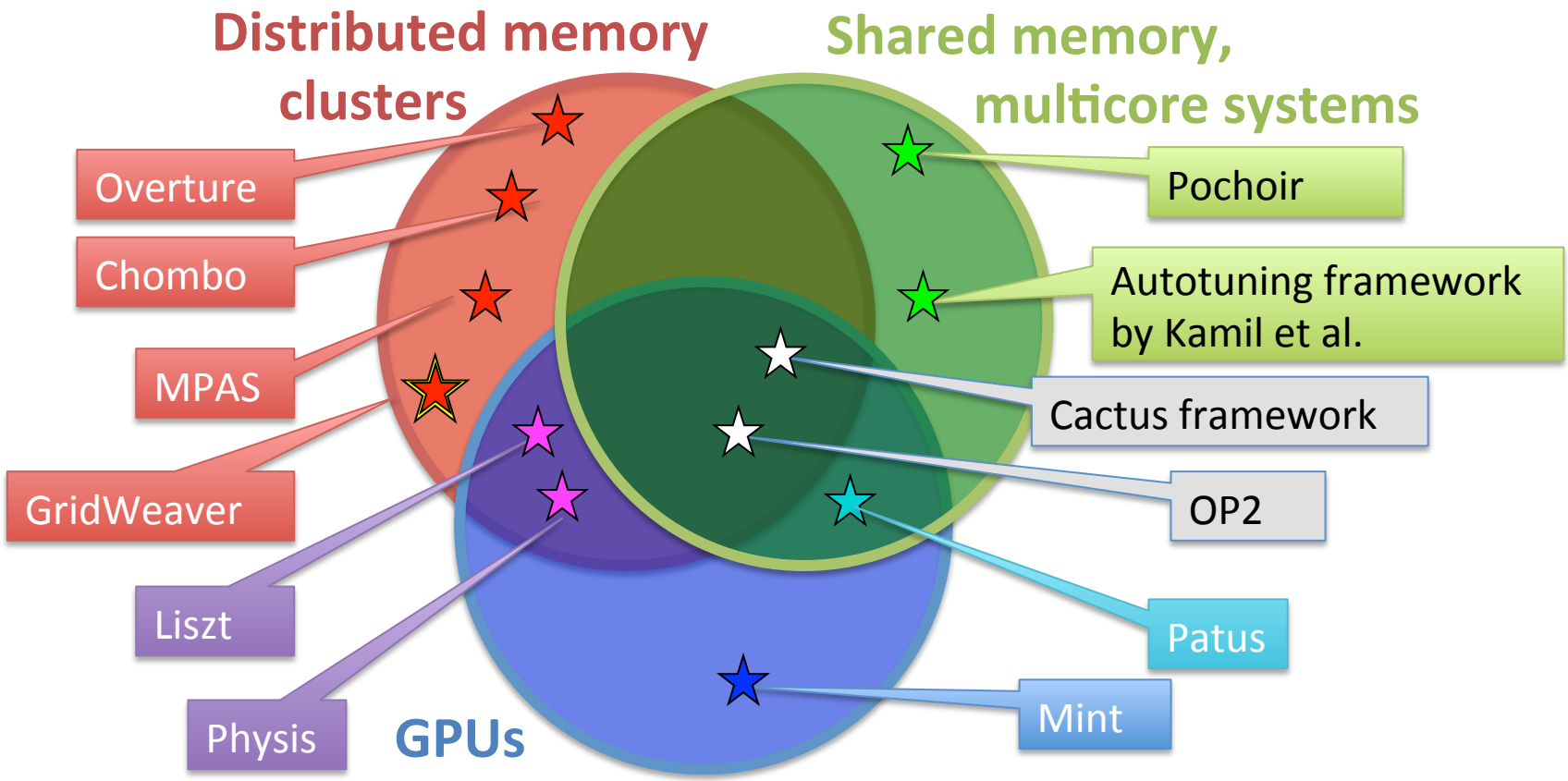
Communication

Conclusions

Publication Record

Programming model approaches

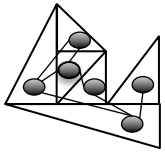
- Programming models are conceptual frameworks that enable programmers to describe and reason about computation.
- Broader term than programming language because language implies a particular syntax.
- Several different models exist for stencil computations on parallel architectures.



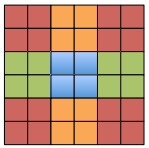
What makes GridWeaver unique?

We can describe and compare models by identifying how they address an application's implementation concerns.

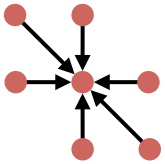
Implementation concerns in stencil computations:



- **Grid connectivity**
- **Data structure for grid values**



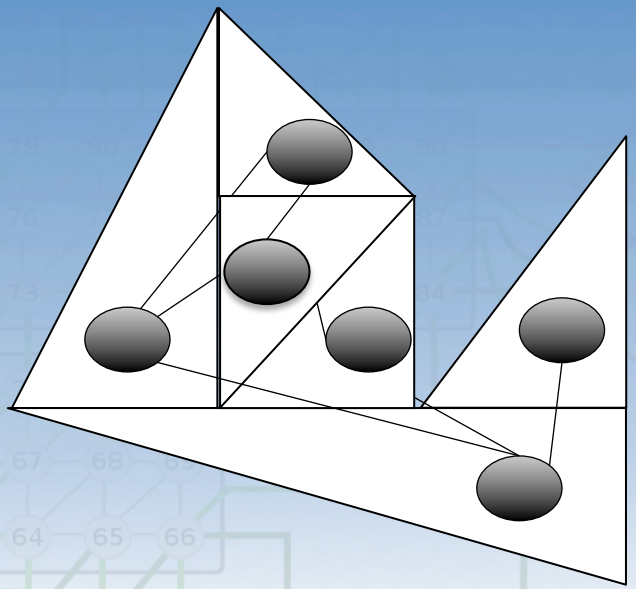
- **Data distribution**
- **Computation distribution**

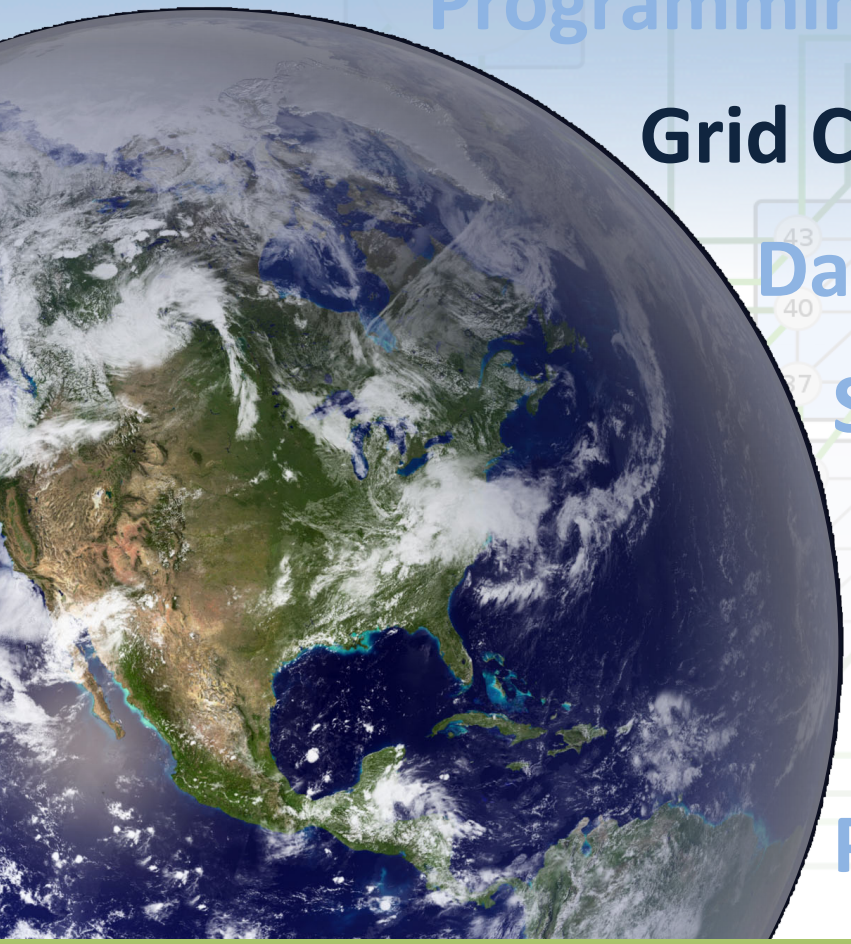


- **Stencil computation**
- **Iteration order**

- **Communication**

Outline





Programming Models

Grid Connectivity

Data\Computation Distribution

Stencil Computations

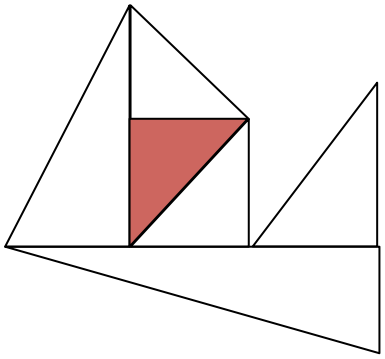
Communication

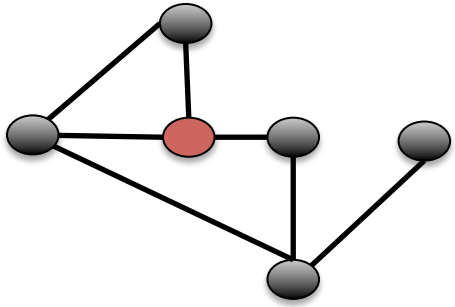
Conclusions

Publication Record

Grid type tradeoffs

Irregular





✔ Most general.

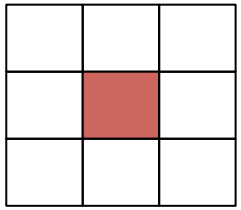
Structured with a graph.

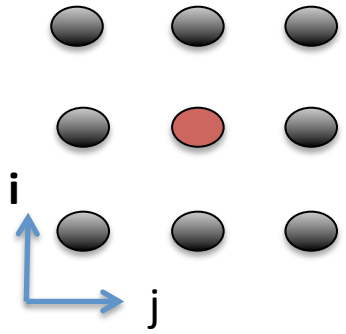
✗ Connectivity is explicitly stored.

✗ Neighbors accessed indirectly through adjacency list.

Regular

✗ Restricted to a regular tessellation of an underlying geometry.





Structured with an array.

✔ Storage efficient.

✔ Neighbors accessed directly in array.

Performance impact of indirect access

```
int direct() {
  int sum = 0;

  for(int i = 0;
      i < DATA_SIZE; i++)
  {
    sum += A[i];
  }

  return sum;
}
```

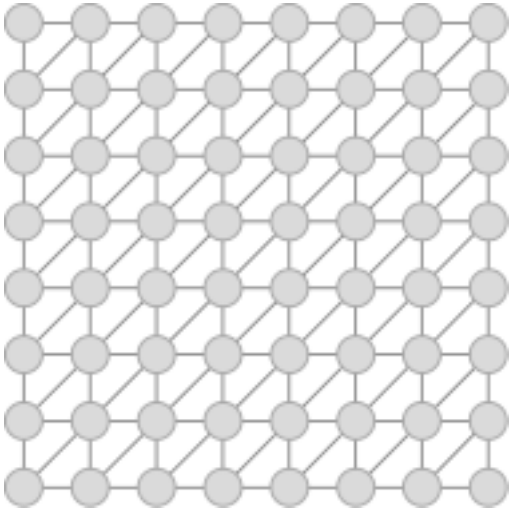
3.47 secs

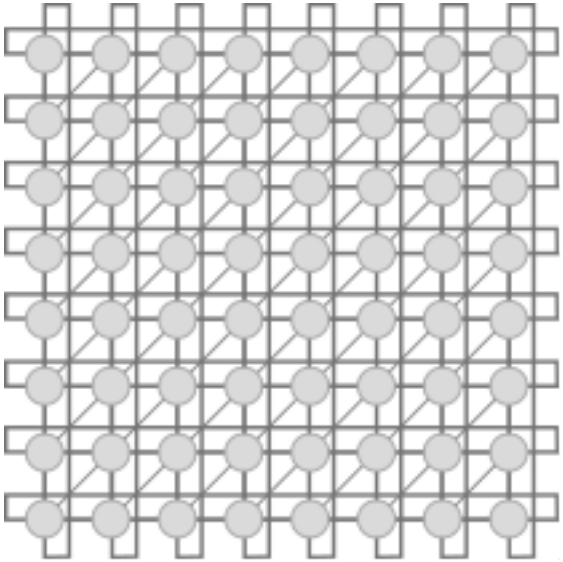
```
int indirect() {
  int sum = 0;

  for(int i = 0;
      i < DATA_SIZE; i++)
  {
    sum += A[B[i]];
  }

  return sum;
}
```

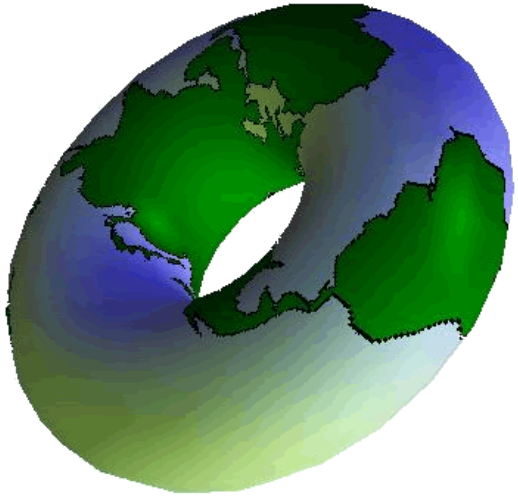
38.41 secs

with DATA_SIZE = 500,000,000 (approx. 1.8 GB)
2.3 GHz Core2Quad machine with 4GB memory

Flexible connectivity matters in Earth Grids

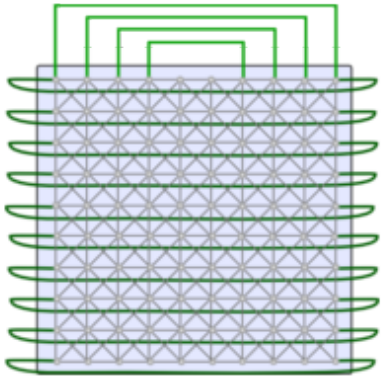




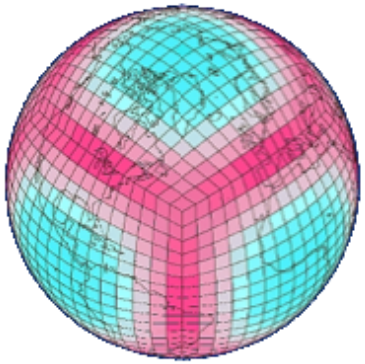


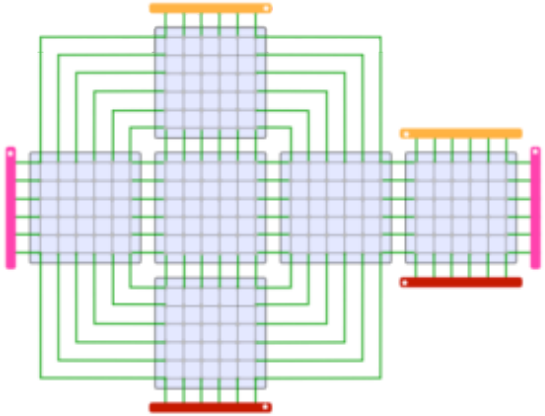
Semiregular Earth grids

Tripole

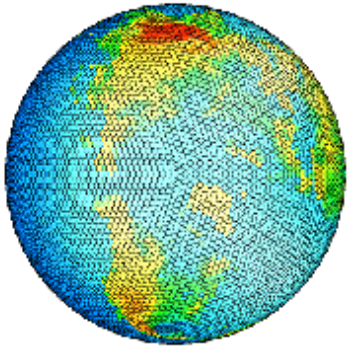


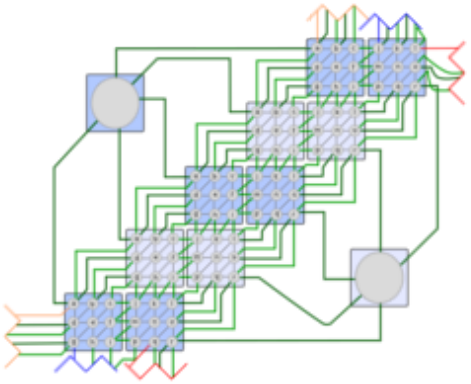
Cubed sphere



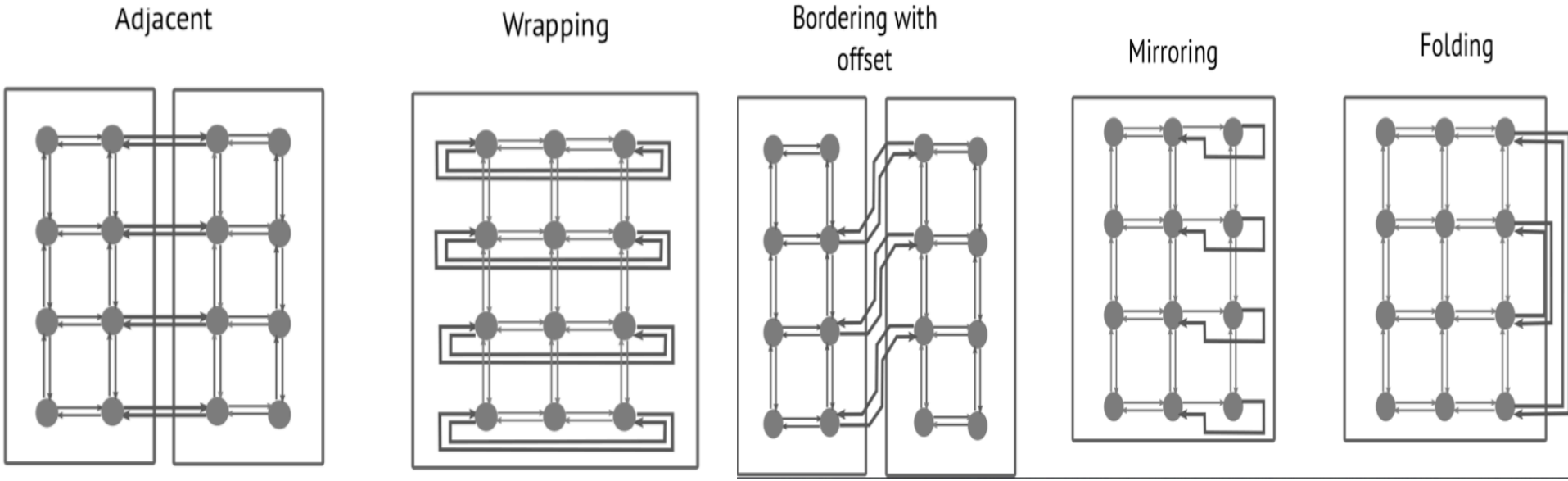


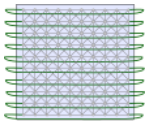
Icosahedral



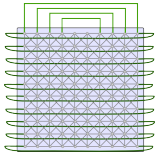


Grid connectivity patterns

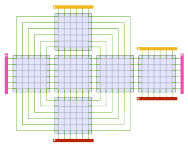




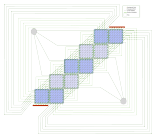
Dipole: Wrapping



Tripole: Wrapping, folding



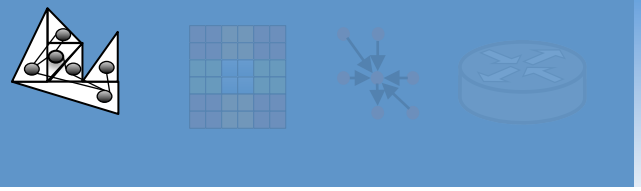
Cubed Sphere: Adjacent



Icosahedral: Adjacent, offset

Expressivity: related work


	Arrays	Graphs	Wrapping	Mirroring	Adjacent	Offset	Folding
ZPL's regions ('98)	✔		✔	✔			
Kamil et. al ('09)	✔		✔				
Mint ('11)	✔		✔				
Patus ('11)	✔		✔				
Physis ('11)	✔		✔				
Pocohir ('11)	✔		✔				
OP2 ('10)		✔	✔	✔	✔	✔	✔
Liszt ('11)		✔	✔	✔	✔	✔	✔
Chombo ('09)	✔		✔	✔	✔	✔	✔
GridWeaver	✔		✔	✔	✔	✔	✔



Expressing semiregular-grid connectivity

Current approach (in SWM and CGPOP):

- Connectivity is not expressed at a high level its hard-coded in communication routines.

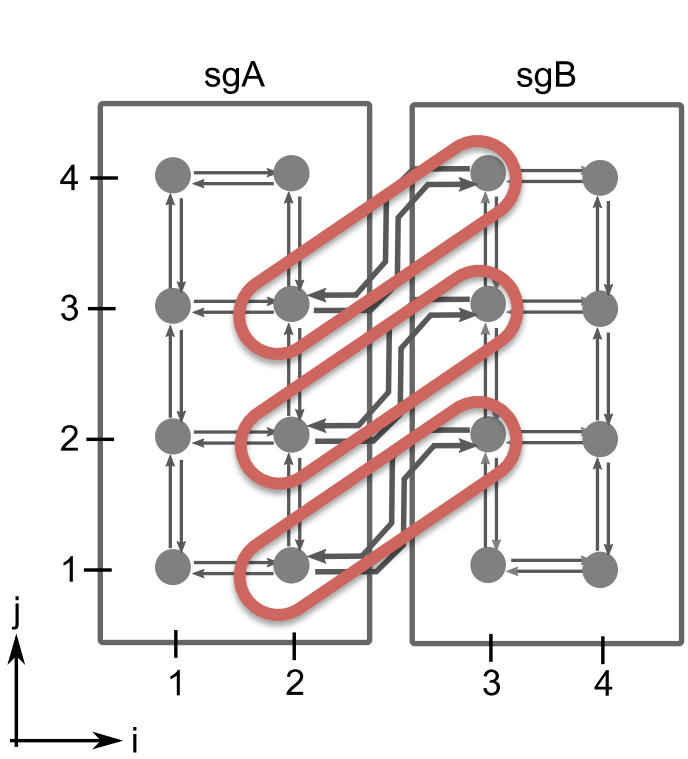
Chombo's approach:

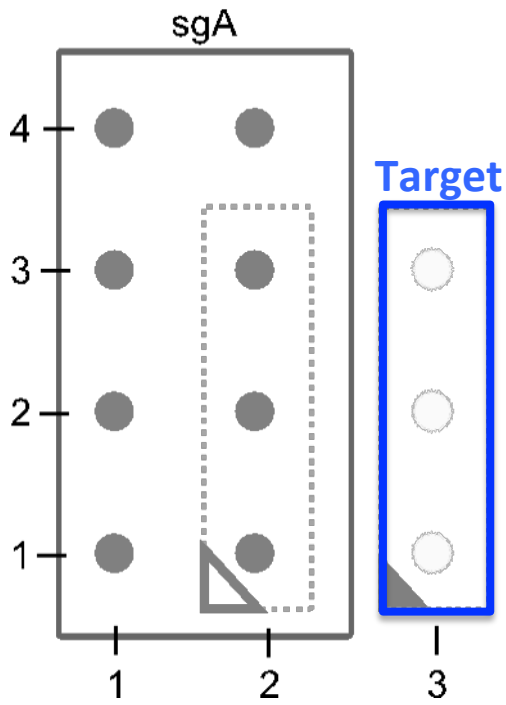
- Connectivity specified by providing equations that map between a 3-dimensional, real, domain and a 2-dimensional, integer, mapped space.
- Chombo is a library for performing computations on adaptively refined meshes.

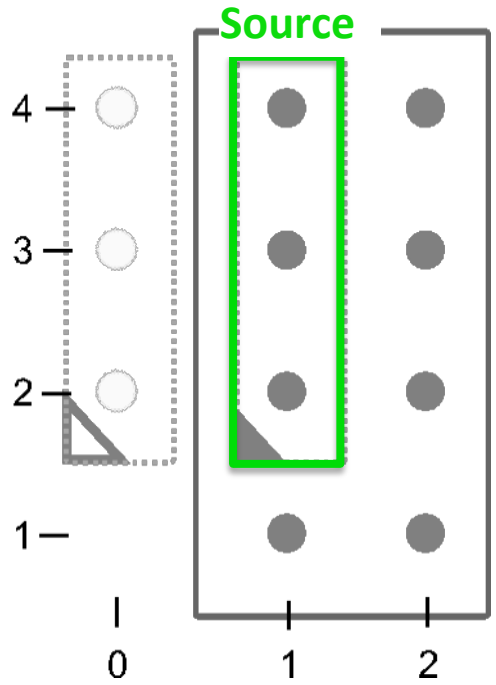
My approach:

- I have developed an abstraction called a border map.

Border maps

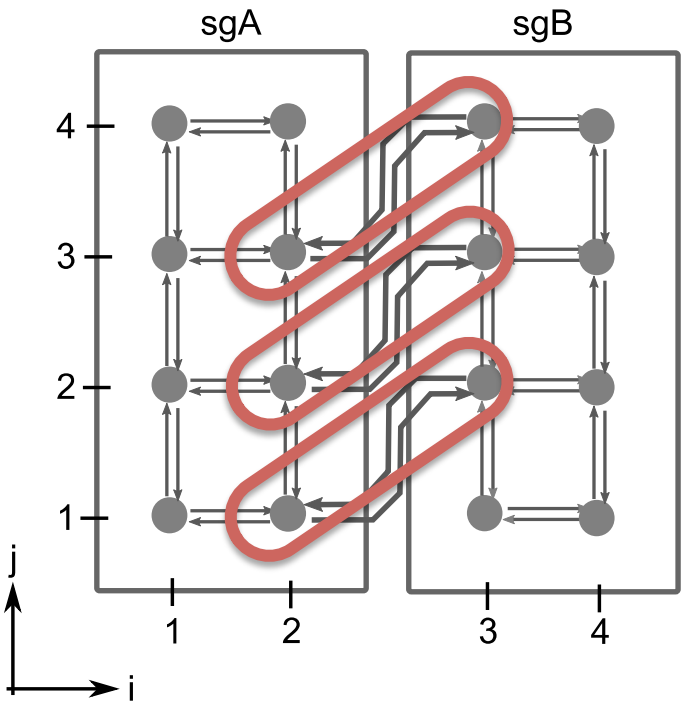


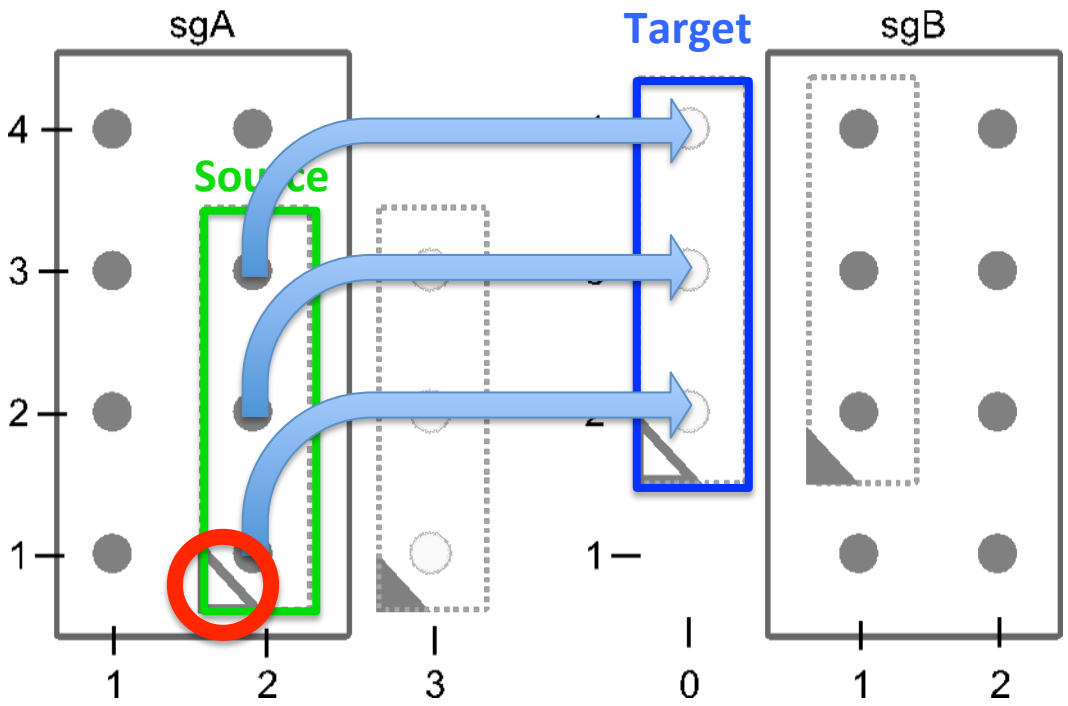




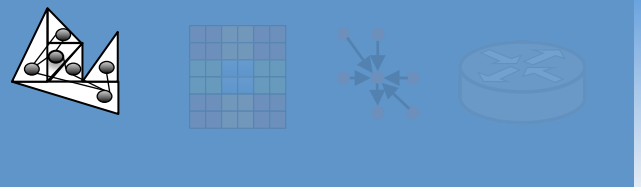

```
call grid_addBorderMap(3,1, 3,3, sgA,
                       1,2, 1,4, sgB)
call grid_addBorderMap(0,2, 0,4, sgB,
                       2,1, 2,3, sgA)
```

Border maps






```
call grid_addBorderMap(3,1, 3,3, sgA,
                       1,2, 1,4, sgB)
call grid_addBorderMap(0,2, 0,4, sgB,
                       2,1, 2,3, sgA)
```



Evaluating how GridWeaver addresses grid connectivity

Expressivity:

- Able to express semiregular grids with border-map abstraction.

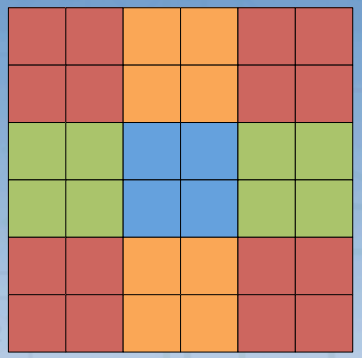
Performance:

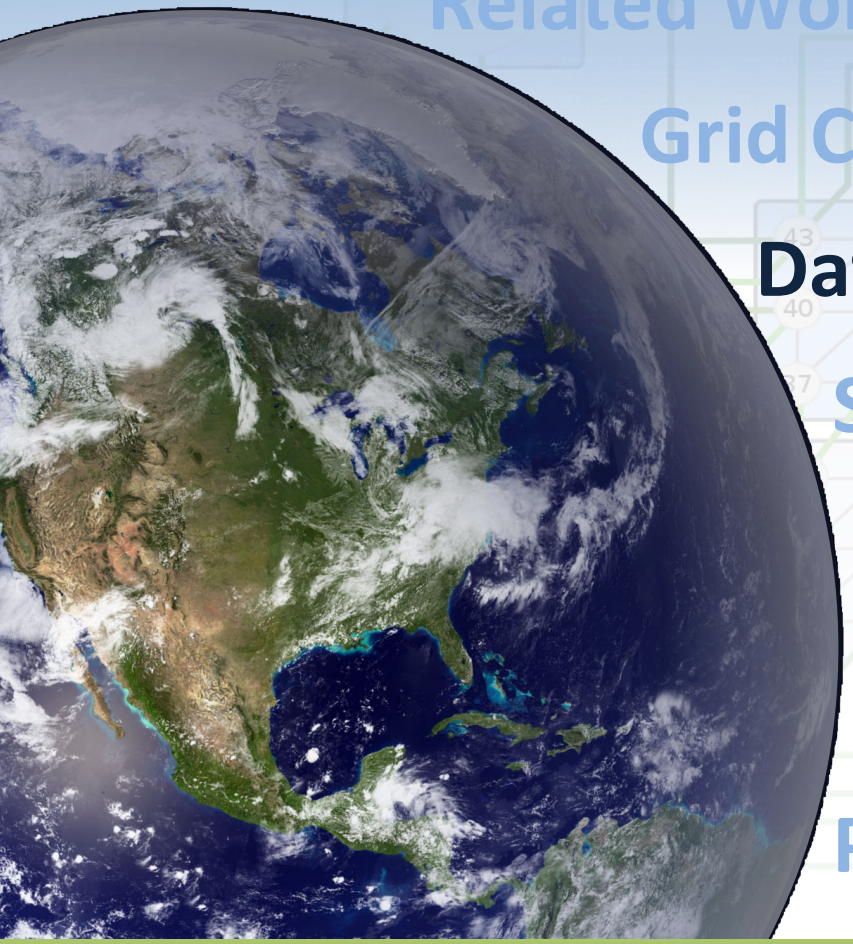
- Data is stored in arrays.

Programmability:

- We enable a separated specification of connectivity.
- Programmers do not have to write any communication code.

Outline





Related Work

Grid Connectivity

Data\Computation Distribution

Stencil Computations

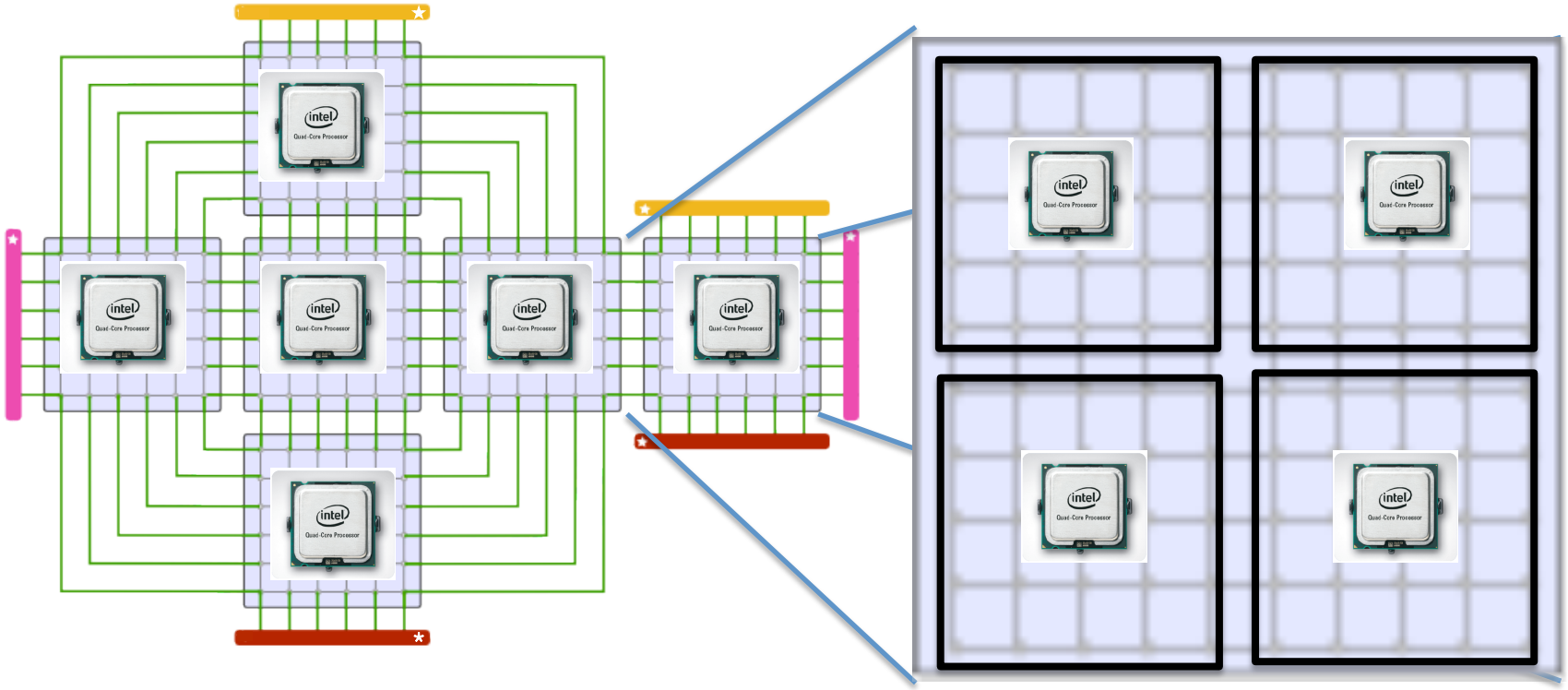
Communication

Conclusions

Publication History

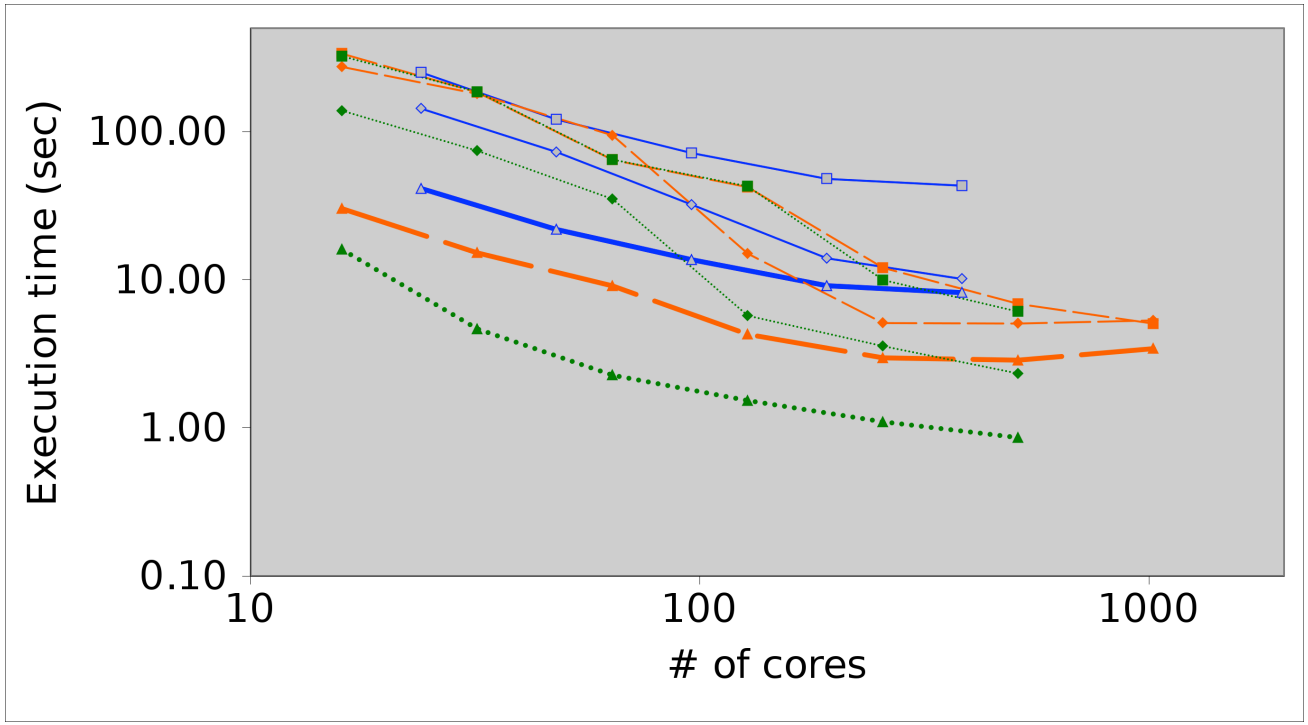
Decomposition and distribution

How to map data and computation onto processors?



Why give users control over distribution?

Time to conduct 10,000 communication calls
on different machines with different distributions



ISTeC Cray
- block-fill
- fill-block
- block-cyclic

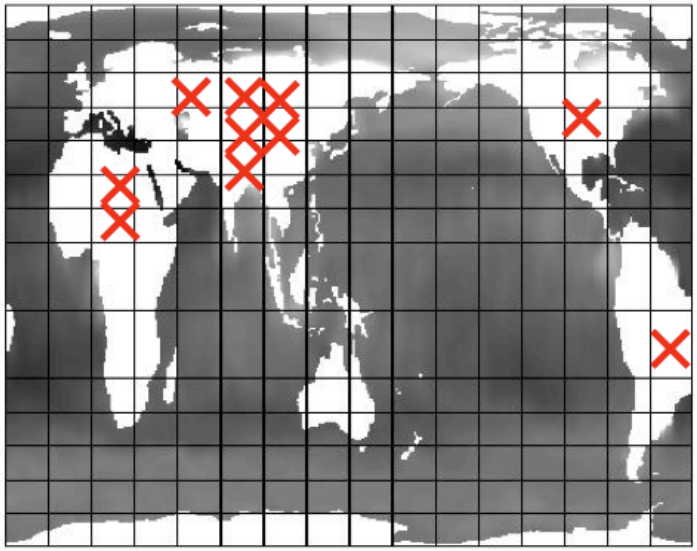
Yellowstone
- block-fill
- fill-block
- block-cyclic

Gordon
- block-fill
- fill-block
- block-cyclic

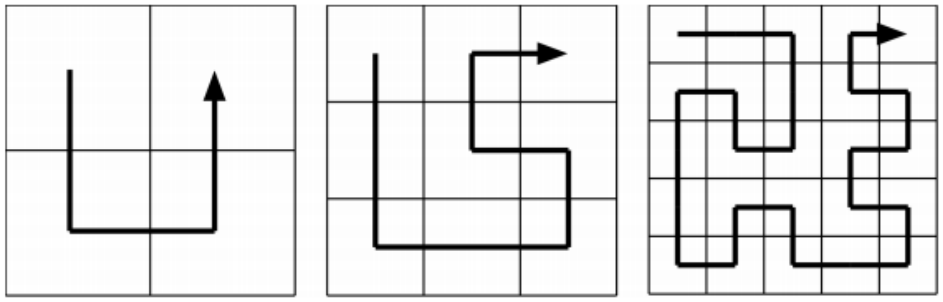
- Performance is highly dependent on the distribution used.
- Therefore its important that GridWeaver be able to run on the machines used for real-world codes and use the same types of distributions seen in real-world codes.

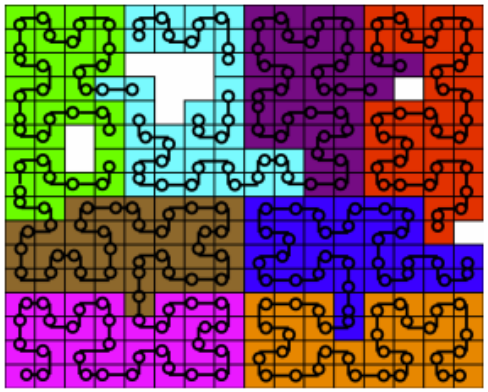
An example distribution: POP's space filling curve

Remove land blocks



Space filling curve



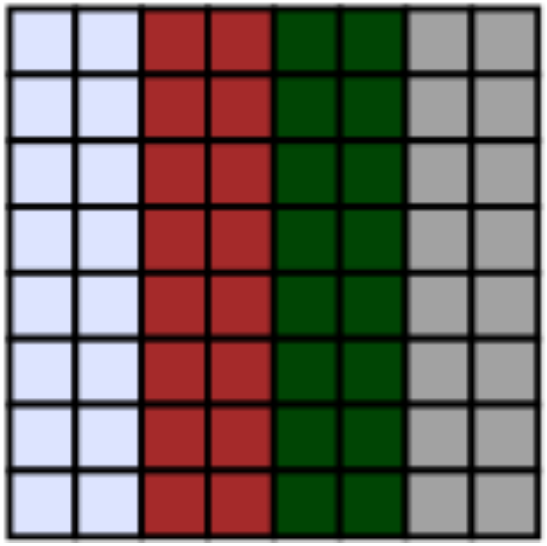


Pictures from: John M. Dennis, "Inverse Space-Filling Curve Partitioning of a GlobalOcean Model," Parallel and Distributed Processing Symposium, International, p. 25, 2007 IEEE International Parallel and Distributed Processing Symposium, 2007

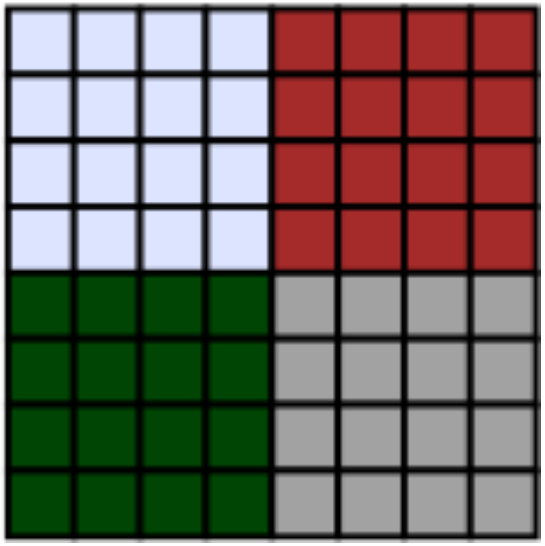
Decomposition and distribution

Fill-Block

Block-Fill

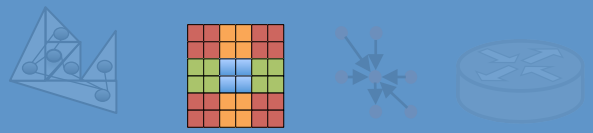


Block-Block



Example:

```
call grid_distributeFillBlock(g, 2)
```



Evaluating how GridWeaver addresses data/computation distribution

Expressivity

- **Computation distribution**
  - Always owner-computes.
- **Data distribution**
  - Single function call for common distributions.
  - Users can explicitly specify mapping for less common / more complicated distributions..
  - We are able to express CGPOP's and SWM's distributions.

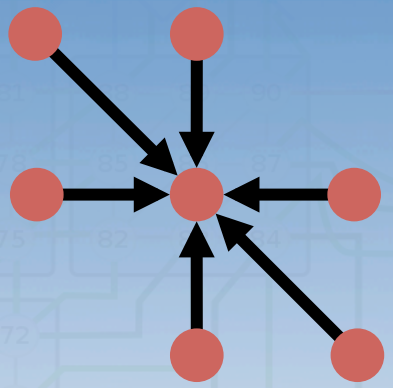
Performance

- Affected by distribution; so it is important that we use distributions seen in real-world codes.

Programmability

- This is the only portion where users have to think about parallelism.
- Users write no communication code.

Outline



Related Work

Grid Connectivity

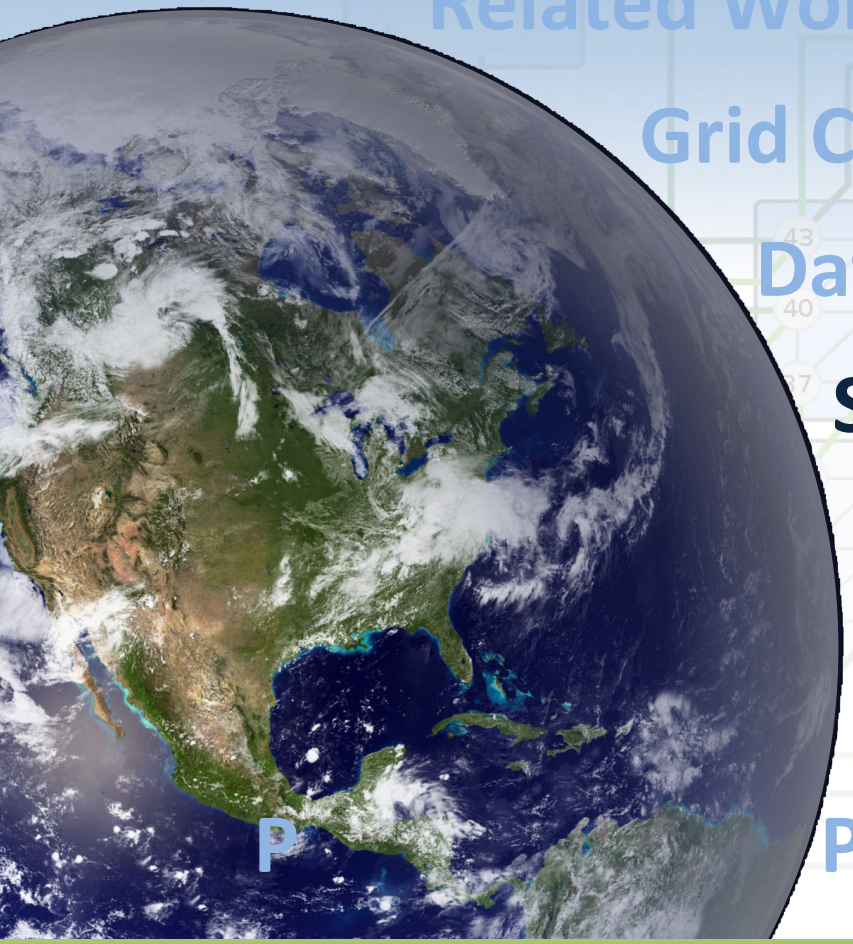
Data\Computation Distribution

Stencil Computations

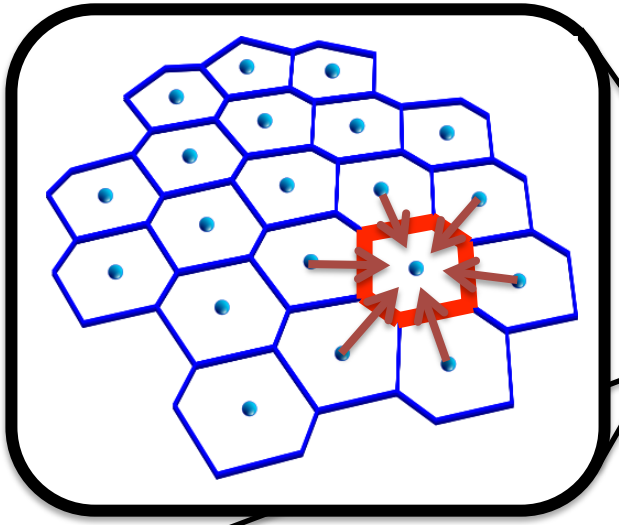
Communication

Conclusions

Publication Record



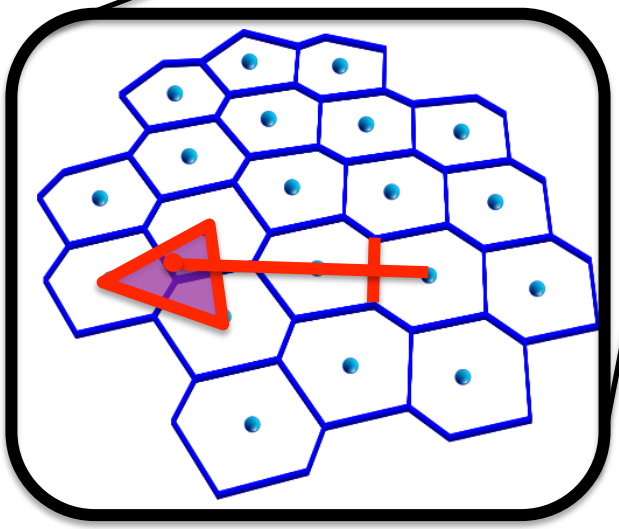
SWM Interpolation point

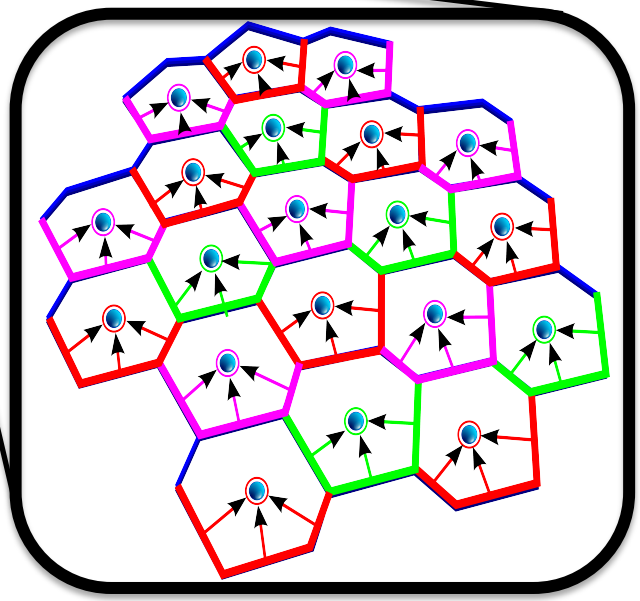


Compact stencil: Access nearest neighbors.
Non-compact stencil: Values further away.

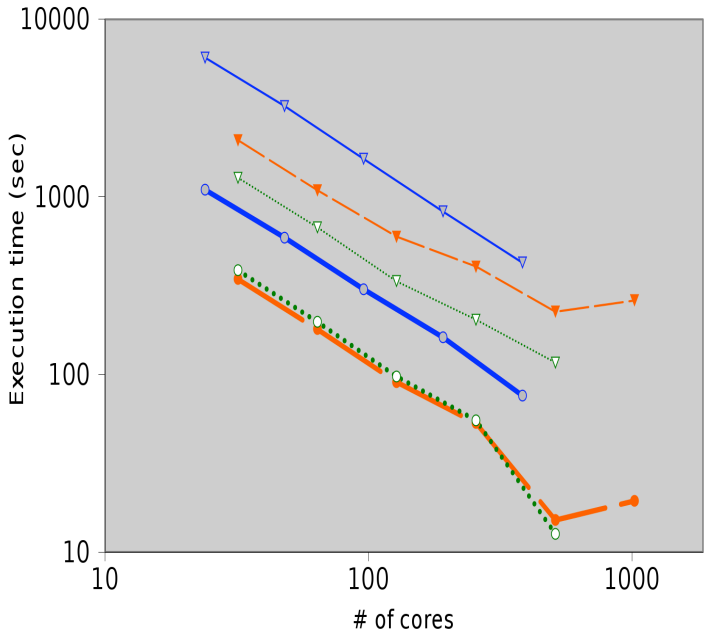
Stencils in SWM:

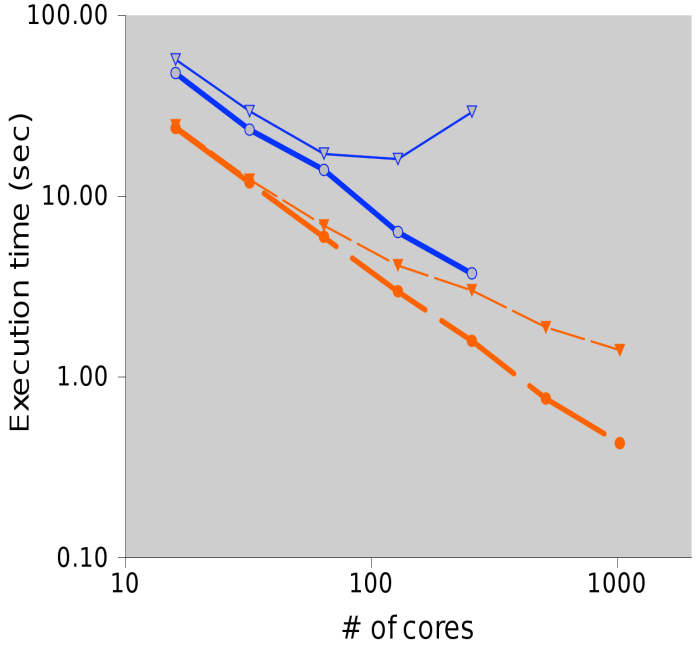
- SWM uses a compact stencil to update cell values using flux on edges.
- SWM uses a non-compact stencil to update edge values.
- We associate three edge values with each cell.

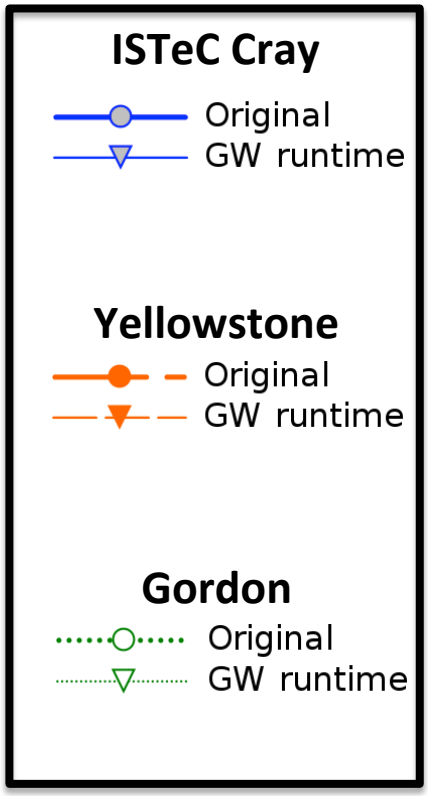




Performance







- Libraries have the advantage of being able to operate with existing code and debugging tools,
- but they introduce library overhead.

Cause of overhead

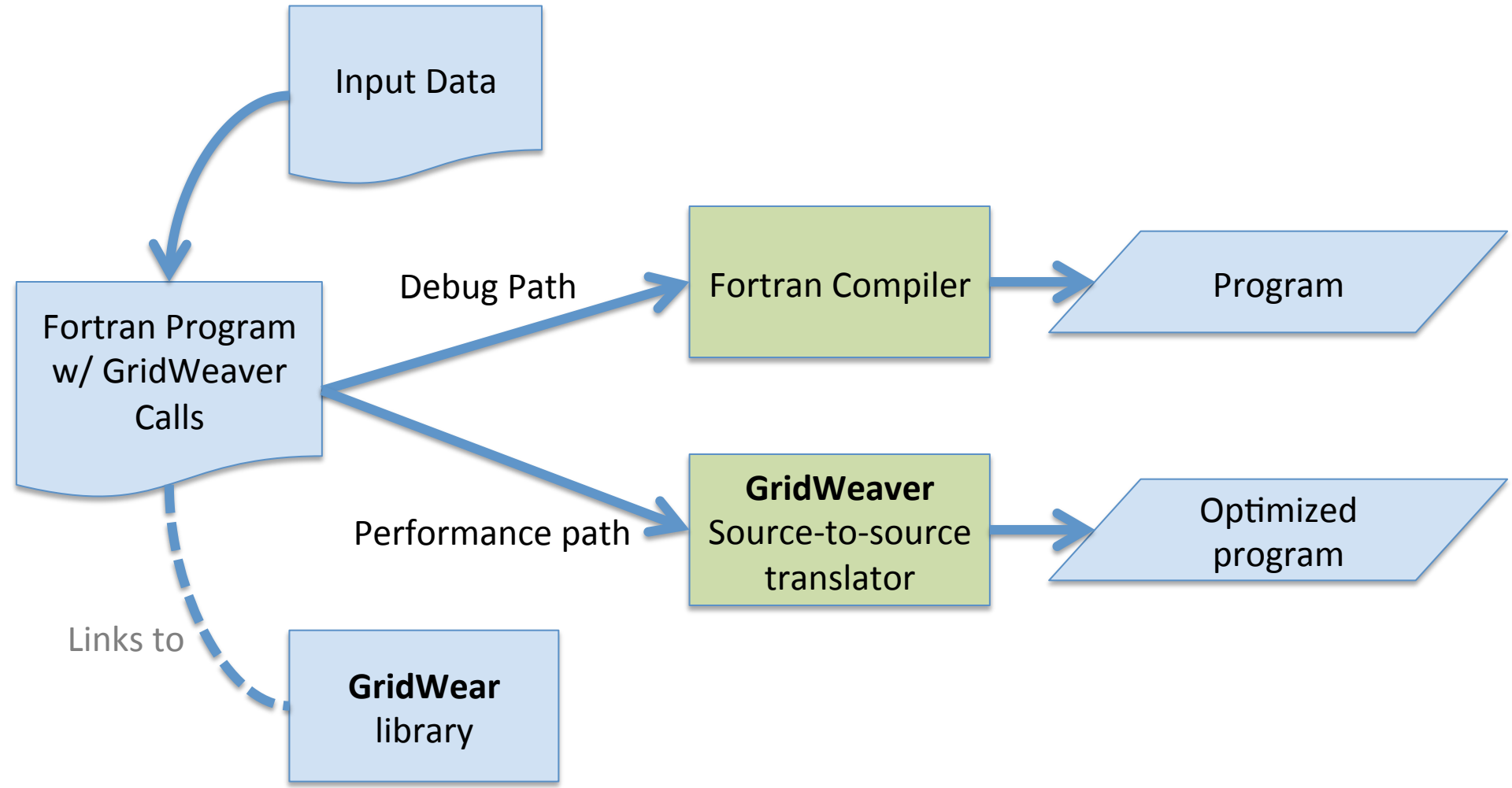
- The GridWeaver library introduces overhead every time it calls a function.
- GridWeaver calls a function:
  - Each time it iterates over a cell (stencil function).
  - Every time the stencil references a node (access function).

```
function fivePt(A, i, j)
  integer, intent(in) :: i, j

  interface
    integer function A(x,y)
      integer, intent(in) :: x,y
    end function
  end interface

  fivePt = 0.2 *
    (A(i, j)   + A(i-1, j) +
     A(i+1, j) + A(i, j-1) + A(i, j+1))
end function
```

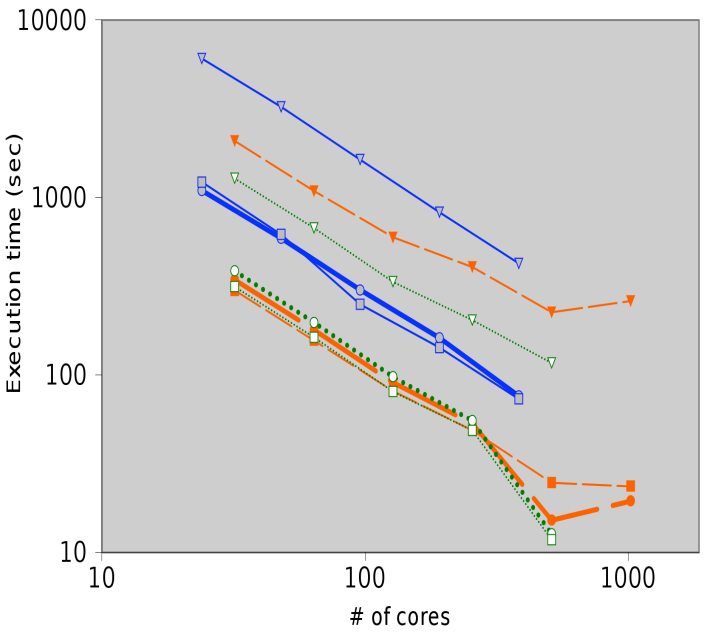
Called once per grid node

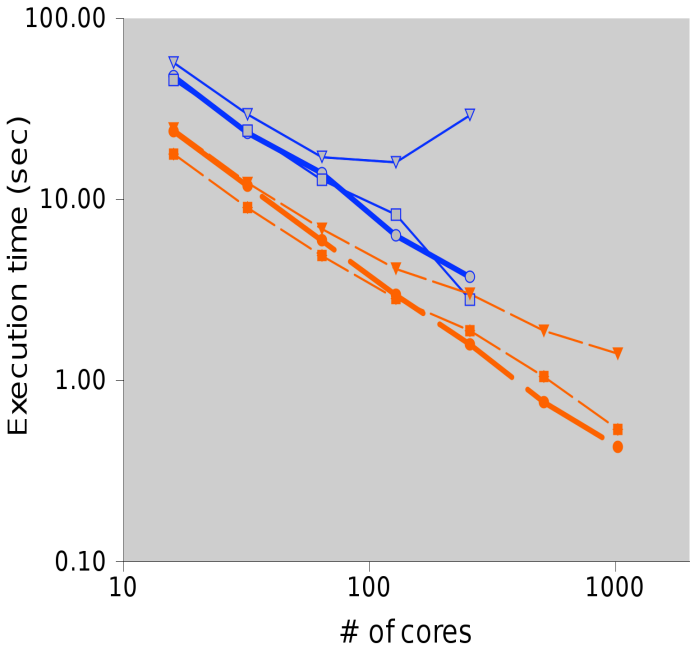
Array-access imposters

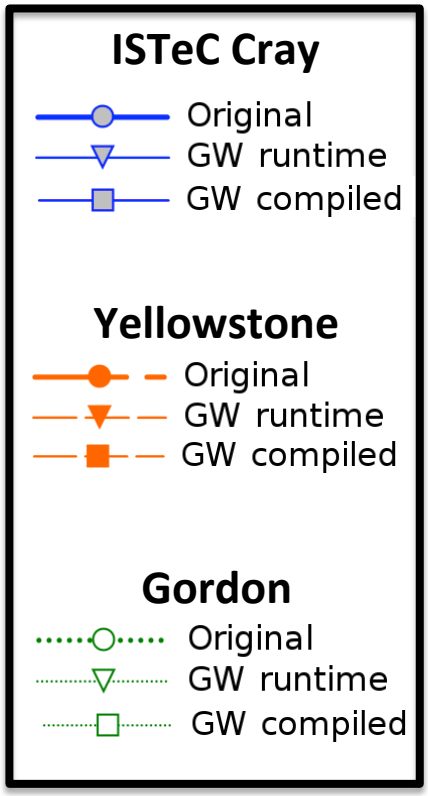
Source-to-source translator



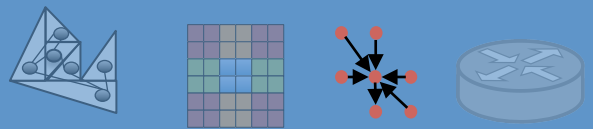
Performance







- The source-to-source translation tool removes library overhead.



Summary: stencils

Expressivity:

- Compact and non-compact stencils within a bounded distance.

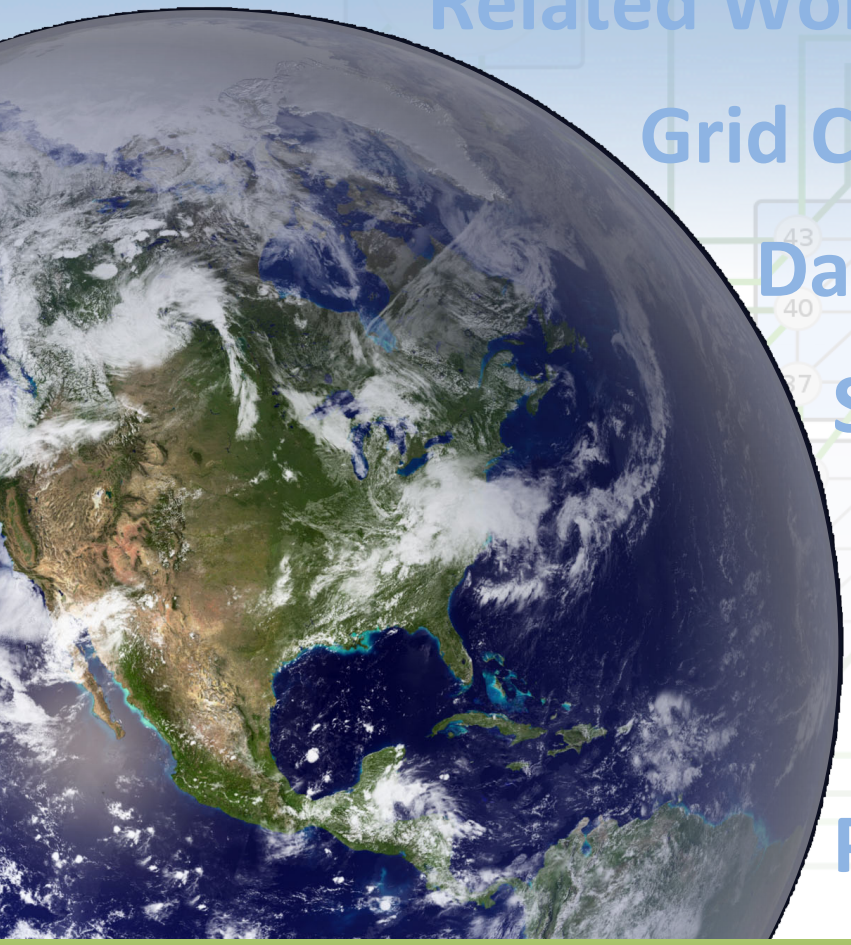
Performance:

- Function call overhead slows things down, but we can alleviate it with a source-to-source translator.

Programmability:

- Able to express stencil computations as if they were operating on a regular grid.

Outline



Related Work

Grid Connectivity

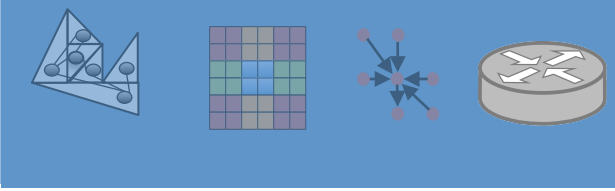
Data\Computation Distribution

Stencil Computations

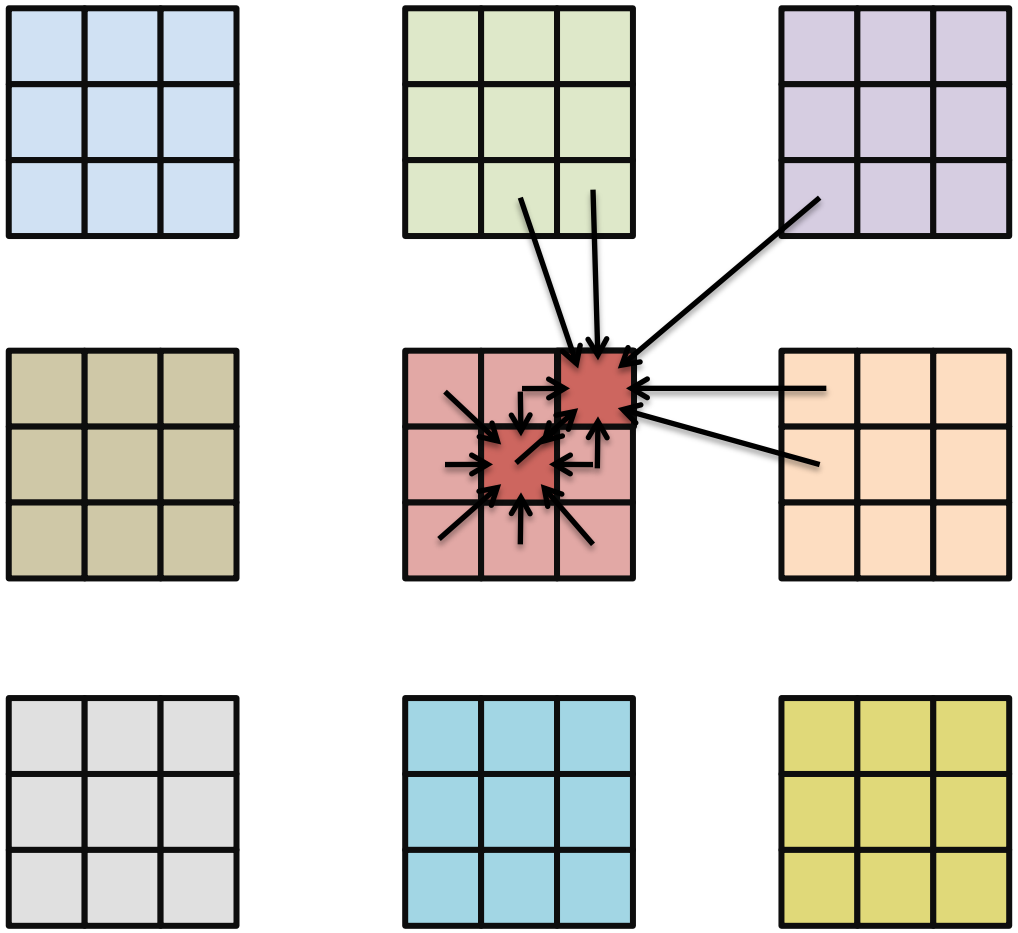
Communication

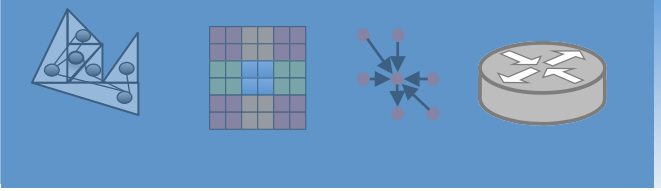
Conclusions

Publication Record

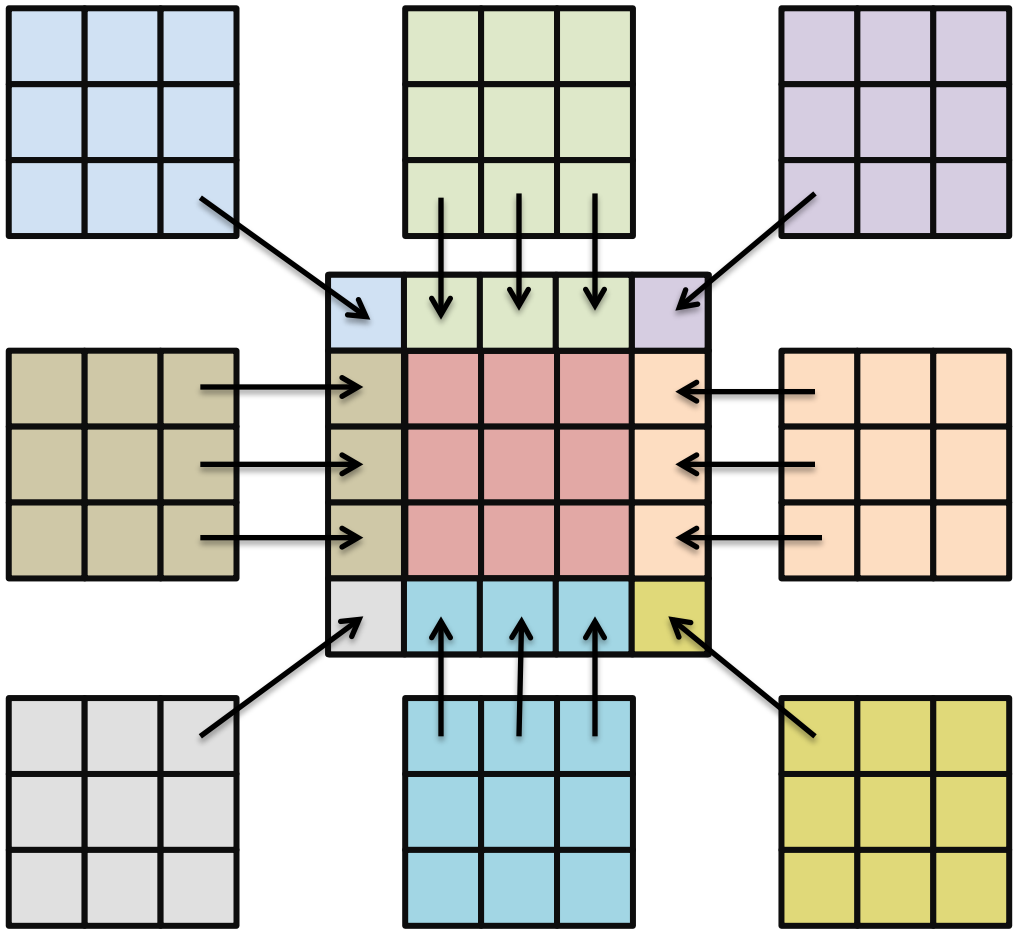


Halos



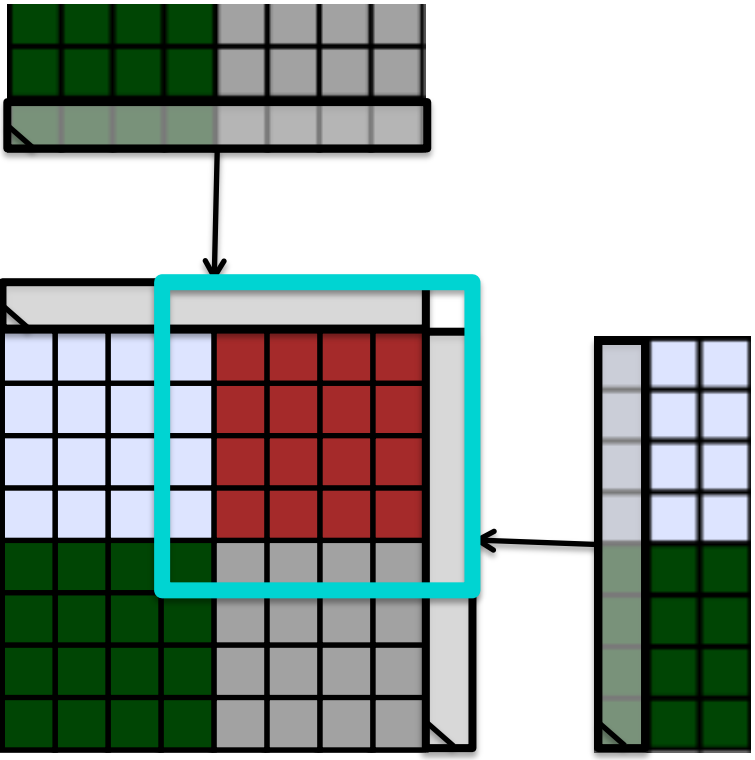


Halos

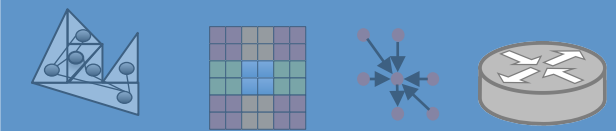


Planning communication

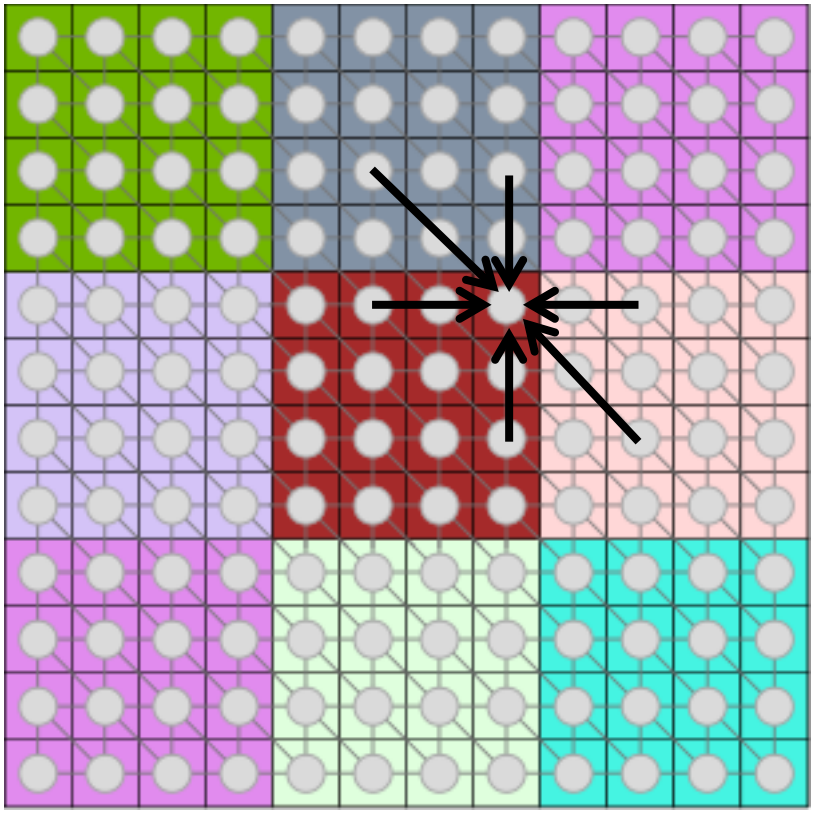
- Before conducting communication generate a communication plan.
  - Communication plan includes:
    - List of Messages to receive and send.
    - What data to send/receive in a message.
    - Who to receive/send a message from/to.

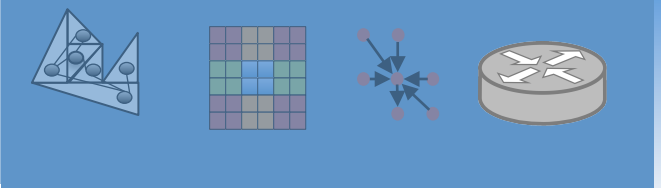


- We determine what communication to conduct by finding out what blocks the halo intersects with.
- If the halo reaches outside the subgrid look at border maps.

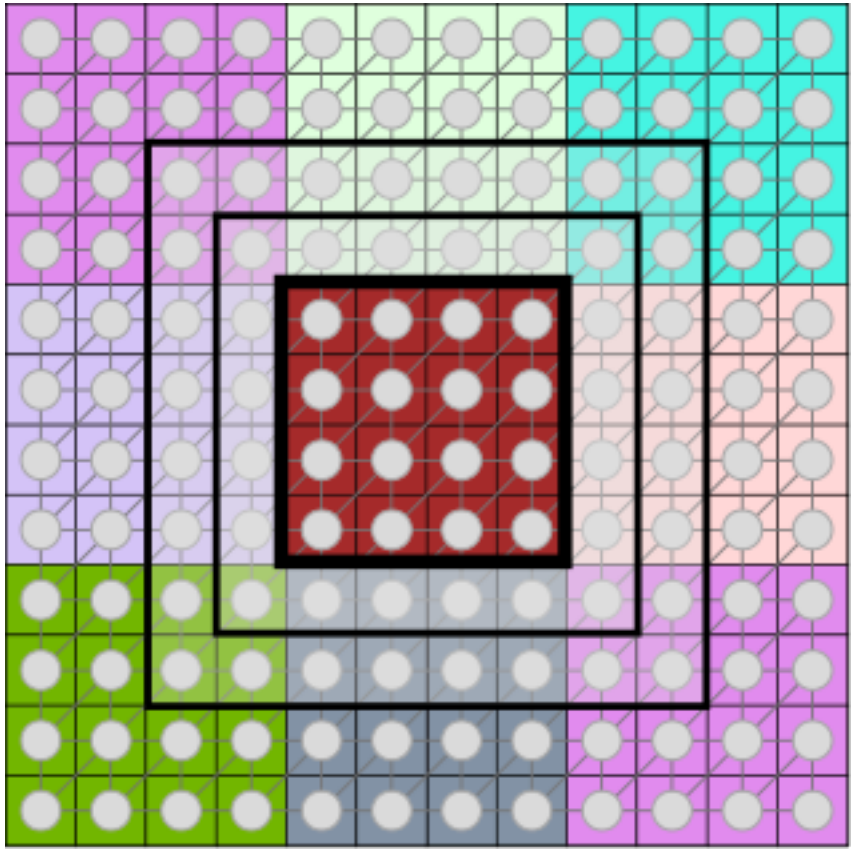


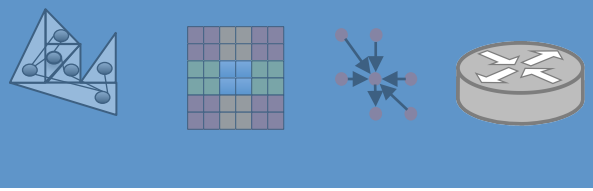
Non-compact stencil



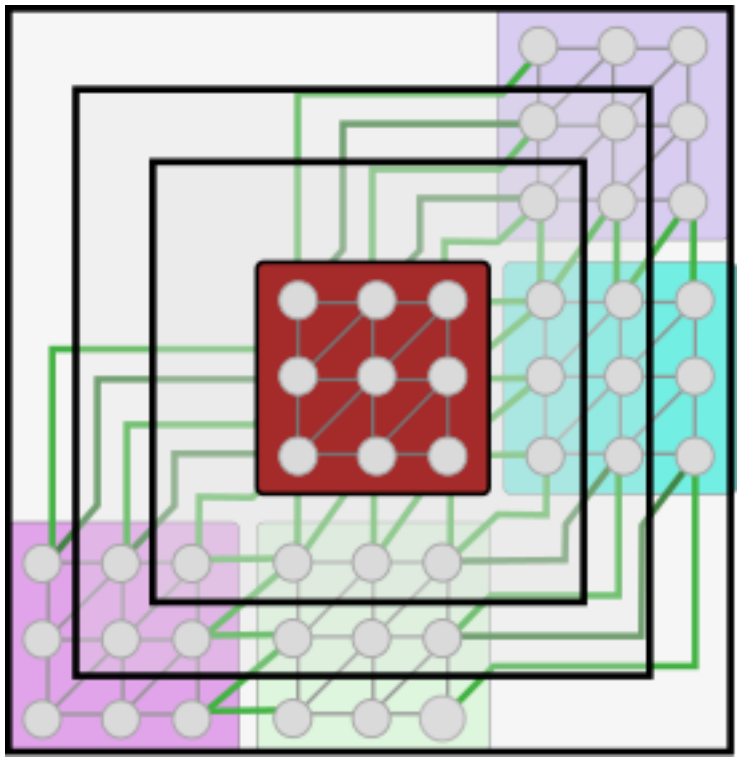


Halos for interior block

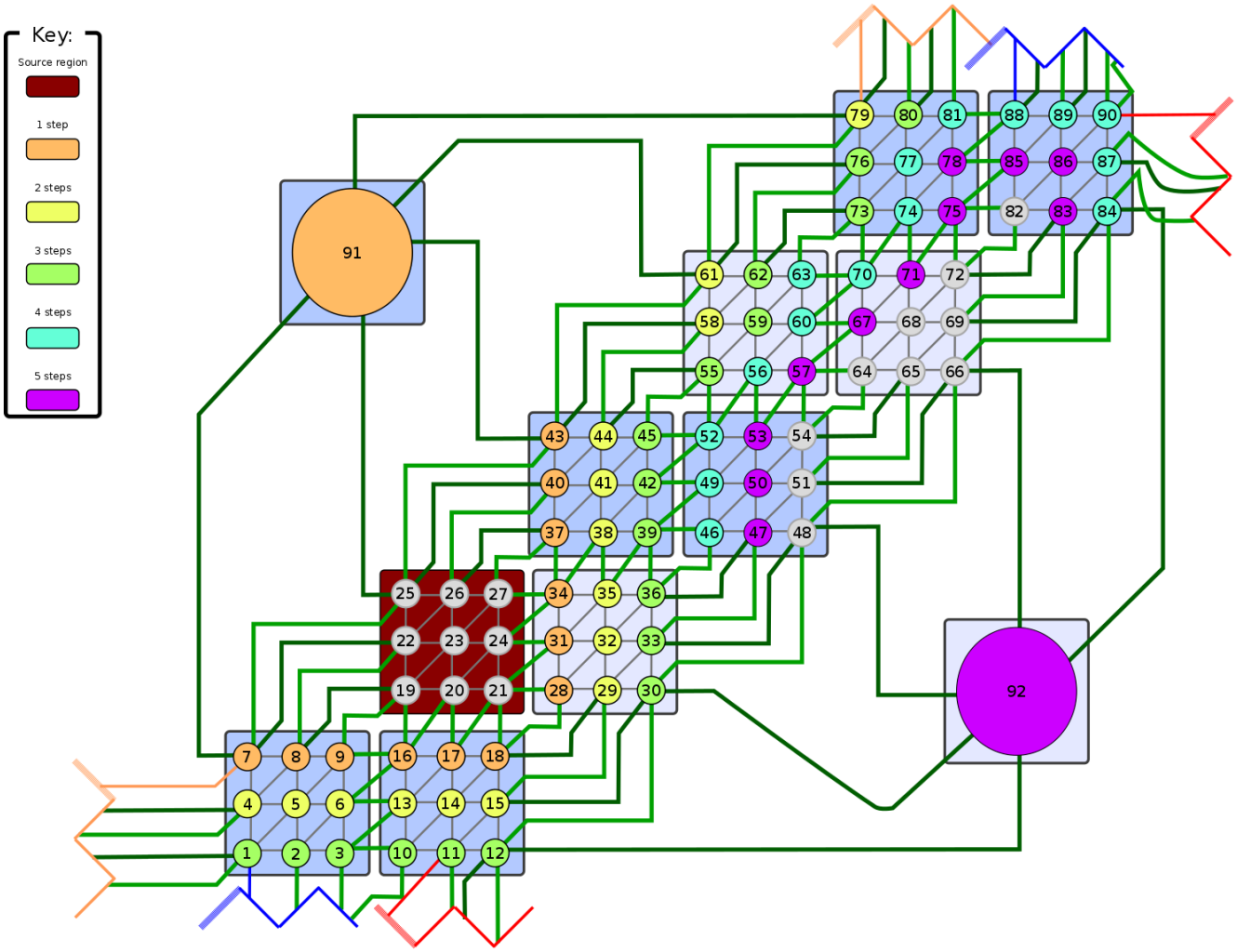




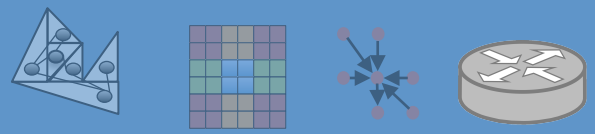
Halos for border blocks?



Ghost-cell lists



- To conduct communication for non-compact stencils do a BFS around a block to see what nodes are accessible.
- Conduct communication to transfer values.
- Store values in sorted list (ghost-cells list), access through an indirection array



Summary: communication

Expressivity:

Communication is invisible to the user. No control but no responsibility.

Performance:

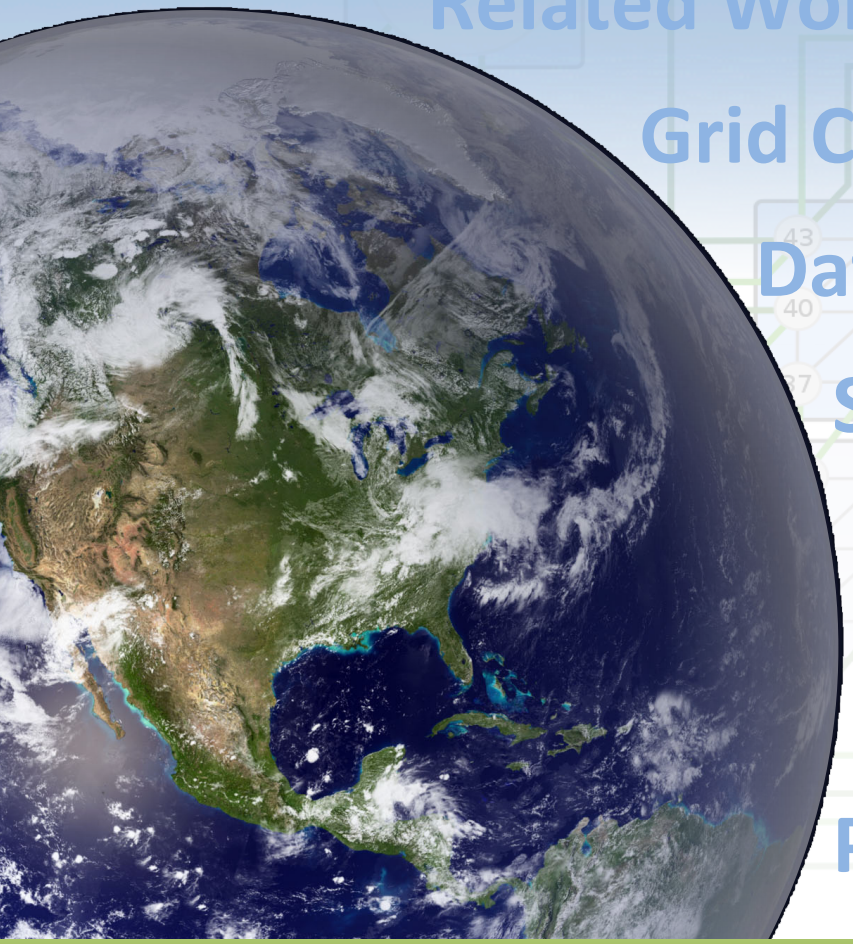
Costs more for larger halos, but difference shrinks as we scale the number of cores up.

GridWeaver automatically aggregates messages.

Programmability:

Communication is invisible to the user. No control but no responsibility.

Outline



Related Work

Grid Connectivity

Data\Computation Distribution

Stencil Computations

Communication

Conclusions

Publication History

Conclusions

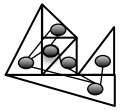
Project goal:

To improve programmability and maintainability of stencil computations in Earth simulation applications.

Approach:

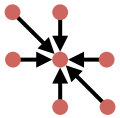
- Identify the implementation concerns inherent in stencils for Earth simulation.
- Separate concerns in a library and match performance with a source-to-source compilation tool.

Implementation concerns:



Grid connectivity . . . . . **Border map abstractions**
Data structure for grid values . . . **Stored in arrays**

Data distribution . . . . . **Functions to map blocks to procs**
Computation distribution . . . . **Owner computes**



Stencil computation . . . . . **User written stencil operator function**
Iteration order . . . . . **Handled by GridWeaver**

Communication . . . . . **Automated by GridWeaver**

Conclusions

Evaluation:

Two case studies:

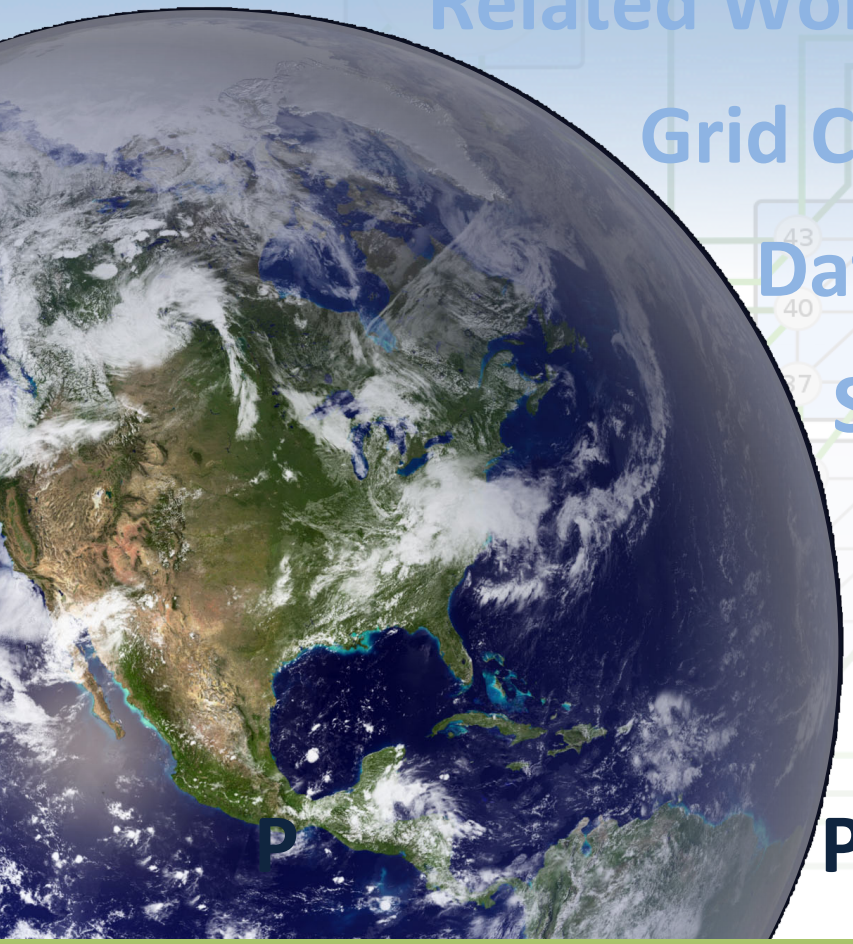
- CGPOP
  - Ocean simulation code.
  - We implement both dipole and tripole versions.
  - Tripole version is the same but with 2 additional lines to specify grid connectivity.
- SWM
  - Icosahedral grid.
  - Non-compact stencil.

We maintain performance by eliminating library overhead by passing code through a source-to-source translation tool

Future Work:

- Additional architectures, grids, and optimizations
- Relaxing implementation limitations
  - Higher dimensionality grids
  - Generic data-types
- Automatically identifying semi-regular portions in existing irregular grids

Outline



Related Work

Grid Connectivity

Data\Computation Distribution

Stencil Computations

Communication

Conclusions

Publication Record

Dissertation work

Programming Abstractions to Separate Concerns in Semi-Regular Grids. **Andrew Stone** and Michelle Mills Strout. Proceedings of the 27th International Conference on Supercomputing (ICS) June 10, 2013.

Evaluating Coarray Fortran with the CGPOP Miniapp. **Andrew I. Stone**, John M. Dennis, and Michelle Mills Strout, Partitioned Global Address Space Conference. October 2011

Mechanisms that Separate Algorithms from Implementations for Parallel Patterns
Christopher D. Krieger and **Andrew Stone** and Michelle Mills Strout. Workshop on Parallel Programming Patterns (ParaPLOP). March 2010

Other work

Automatic Determination of May/Must Set Usage in Data-Flow Analysis. **Andrew Stone** and Michelle Mills Strout and Shweta Behere. International Working Conference on Source Code Analysis and Manipulation (SCAM). September 2008

May/Must Analysis and the DFAGen Data-flow Analysis Generator. **Andrew Stone** and Michelle Mills Strout and Shweta Behere. Information and Software Technology, 51(10) October 2009

Scalable Simulation of Complex Network Routing Policies. **Andrew I. Stone** and Steven DiBenedetto and Michelle Mills Strout and Daniel Massey. The Proceedings of the ACM International Conference on Computing Frontiers (CF) 2010

Acknowledgements

Advisor: Michelle Strout

Commitee:

Daniel Massey

Shirdeep Palickara

David Randell

Miniapp developers \ support:

CGPOP: John Dennis

SWM: Ross Heikes

Machines access \ support:

ISTeC Cray (Colorado State)

XSede Startup Allocation

NCAR University Allocation