Using the Distributed Coupling Toolkit (DCT) to couple model components of Generalized Curvilinear Environmental Model

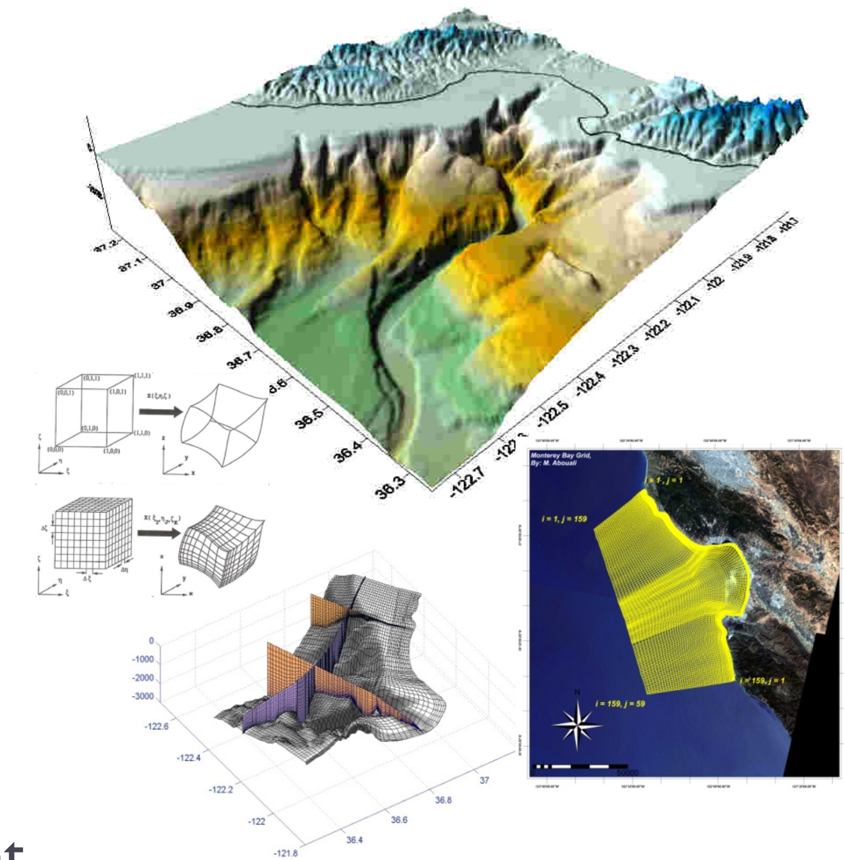
Dany De Cecchis. PhD Candidate.
CSRC. San Diego State University

Outline

- Motivation
- Weak Model Coupling
- Type of Model Coupling
- Coupling in HPC
- DCT
- Applications
- Summary and Future Plans

GCEM

- Motivation
  - Increase the complexity of simulation
  - Combine interaction ocean-atmosphere
  - High resolution coastal simulation
- Objective
  - Develop high resolution and boundary fitting modeling to simulate a coastal environment.



San Diego State University

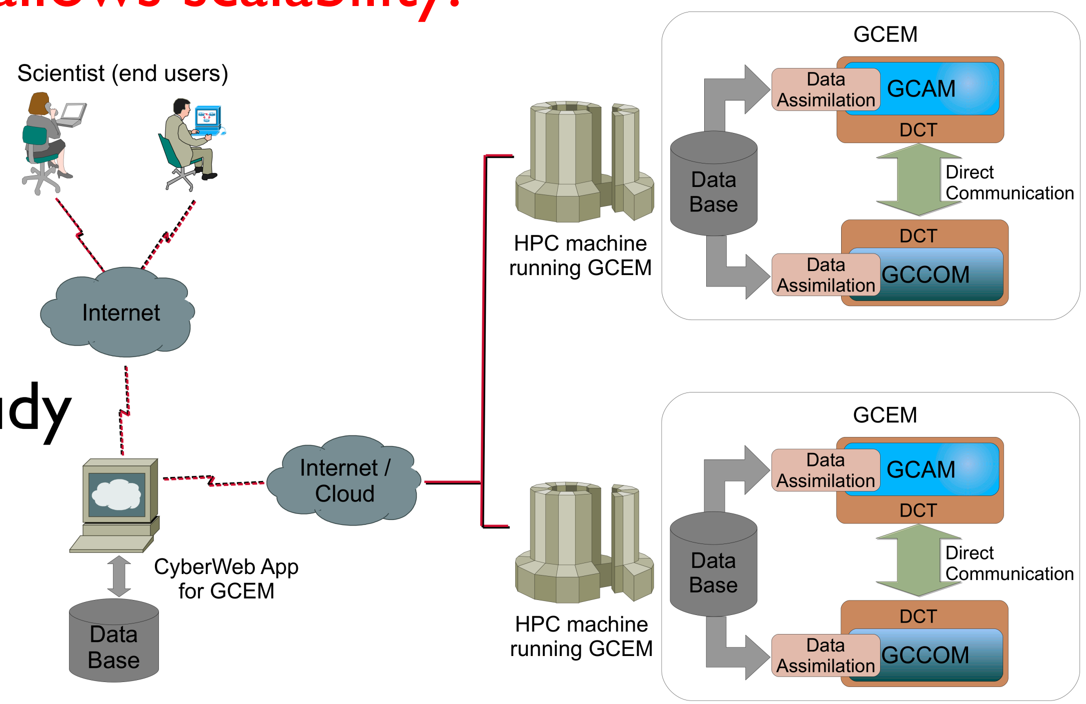
CSRC

Office of Science U.S. Department of Energy

Berkeley Lab

DCT on GCEM

- Goal: Study the problem of coupling different computational models in earth science developing coupling tools that allows scalability.



- A key objective: Study and Implement the coupling of GCEM components using DCT.

Motivation

Why to couple?

- To increase complexity; i.e. obtain more realistic modeling.
- To study interactions between phenomena.
- To overcome model resolution limitations
- To get over complicated mesh generation.

Motivation

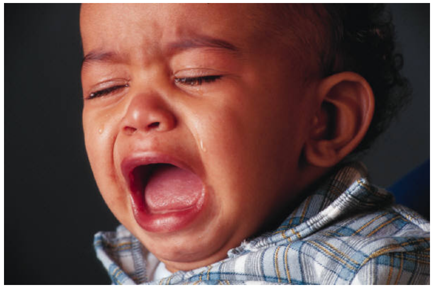
What is coupling?

- Incorporate effects from another natural process:
  - Coupling equations (strong coupling) So complicated!!!

Motivation

What is coupling?

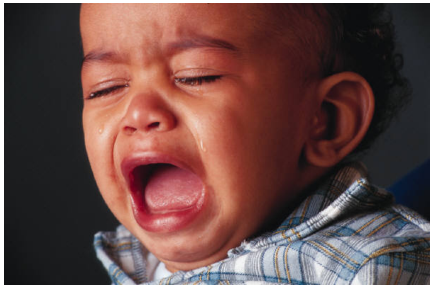
- Incorporate effects from another natural process:
  - Coupling equations (strong coupling) So complicated!!!



Motivation

What is coupling?

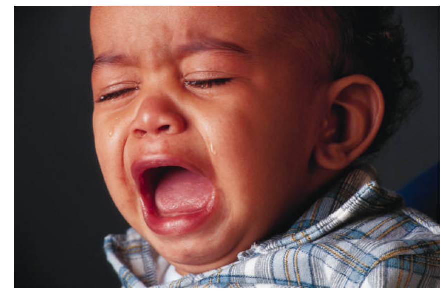
- Incorporate effects from another natural process:
  - Coupling equations (strong coupling) So complicated!!!
  - Providing values as result of another process (weak coupling)
    - Simpler than above one.



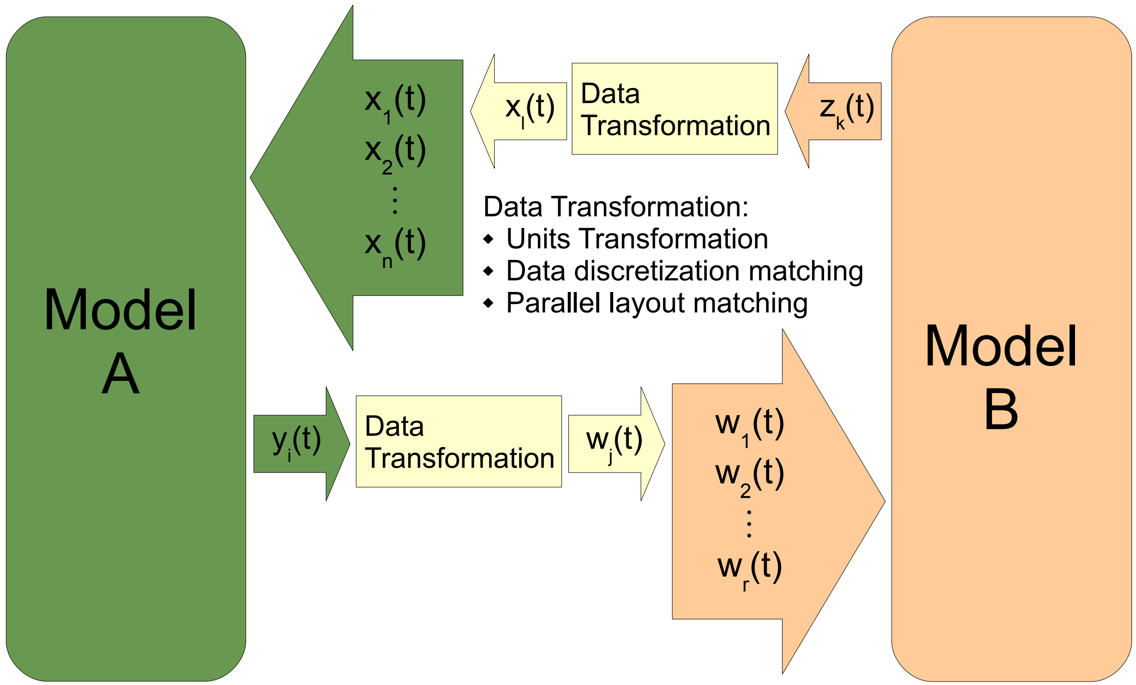
Motivation

What is coupling?

- Incorporate effects from another natural process:
  - Coupling equations (strong coupling) So complicated!!!
  - Providing values as result of another process (weak coupling)
    - Simpler than above one.
    - Could produce instability.
    - Not always a simple process, anyways…



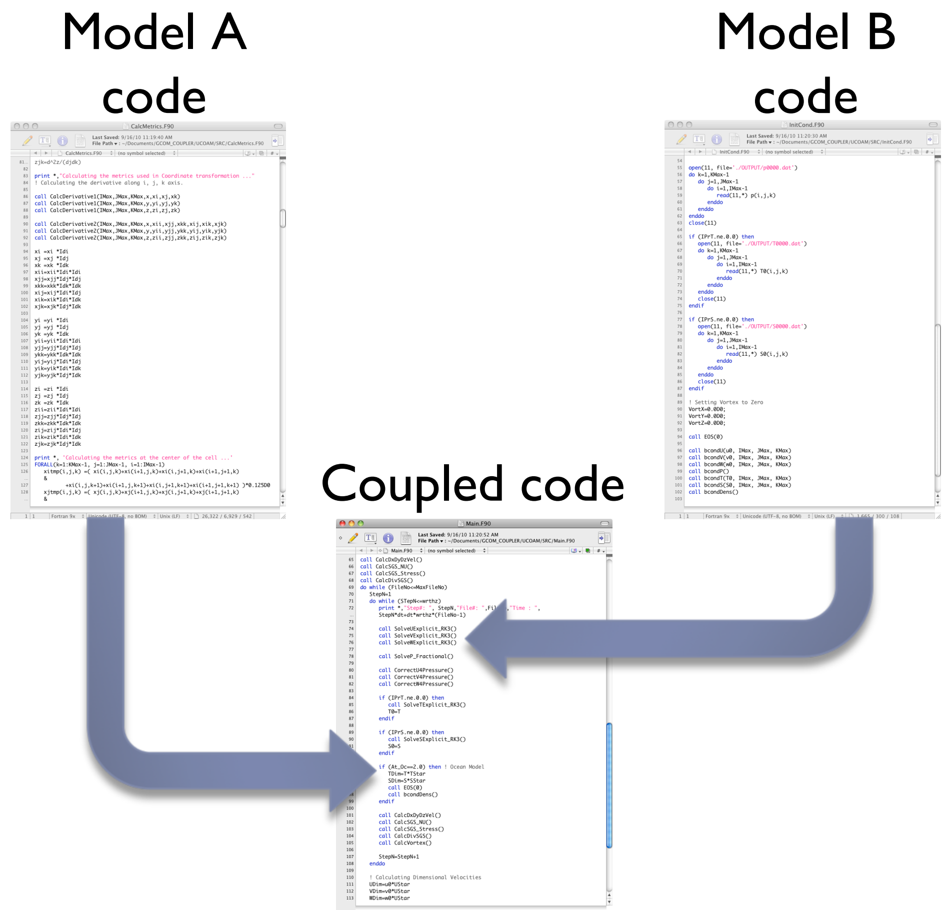
Weak Model Coupling



- Where two models exchange variables or fields values each other.
- It is defined in a common or overlapped domain between both models.
- It happens in determined time intervals.
- The coupling can be made on models running either sequentially or concurrently.
- The exchanged data could require some transformation (filters, interpolation, etc.)

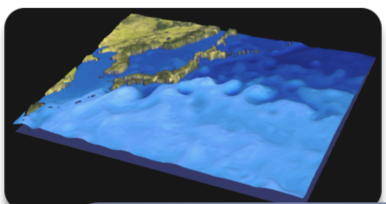
Type of Model Coupling

- **Monolithic code.**
  - Merging model codes in one.
  - Also known as *subroutinization*.
  - Difficult to maintain.
  - Usually sequential and difficult to parallelize.

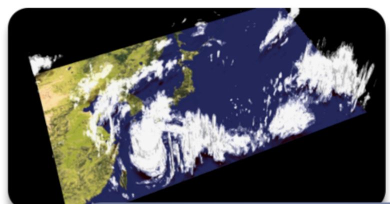


Type of Model Coupling

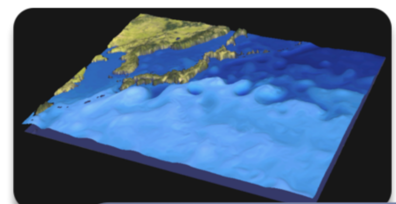
- Scheduled coupling.
  - Schedule model runs using output.
  - Preprocessors perform transformation and filter.
  - Difficult to maintain.
  - Difficult to parallelize.
  - Slow performance.



Model A
- Few time steps
- Output files



Model B
- Preprocess file
- Few time steps
- Output files

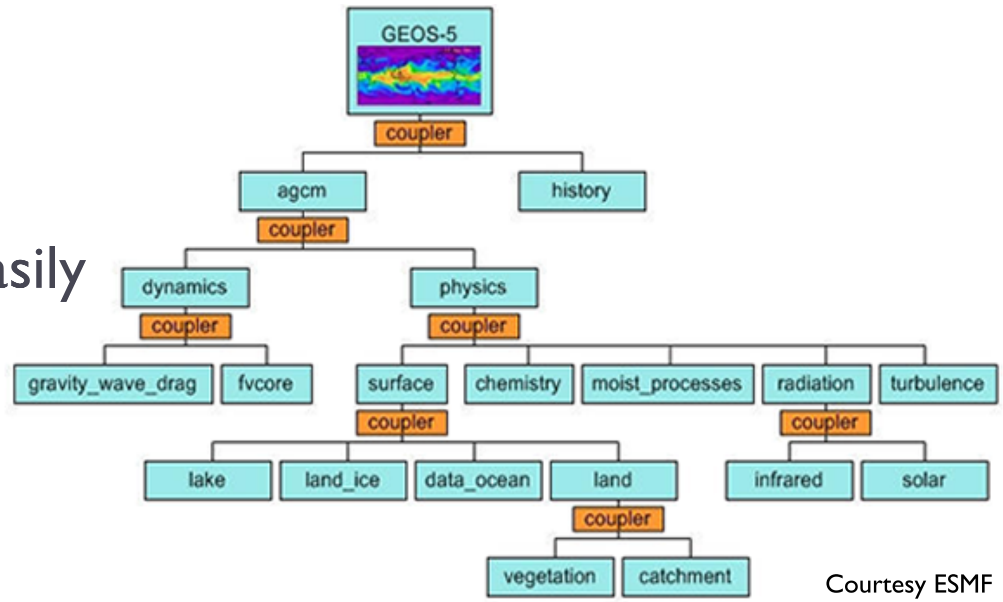


Model A
- Preprocess file
- Few time steps
- Output files

Pictures: Courtesy ESC, Japan Agency for Marine-Earth Science and Technology

Type of Model Coupling

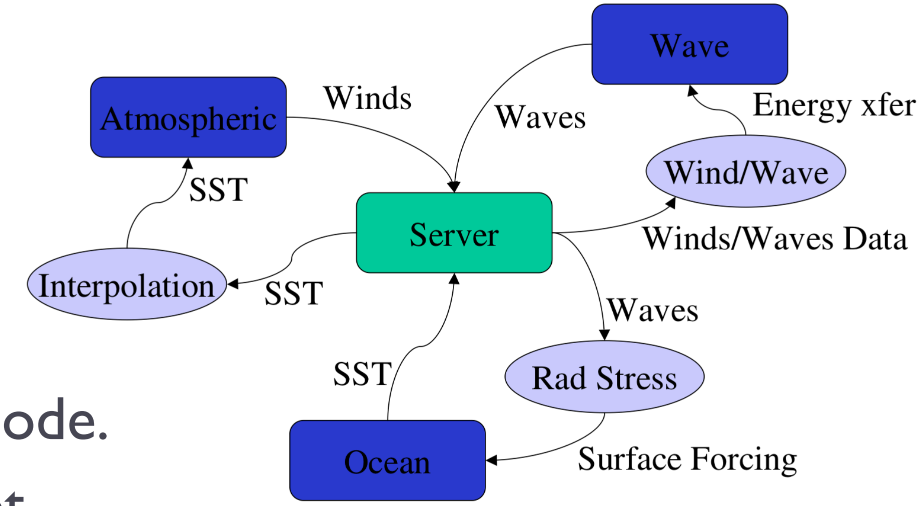
- **Component Approach.**
  - Split in small functional pieces.
  - Need recode existing models.
  - Interface may not be easily separated.



Courtesy ESMF

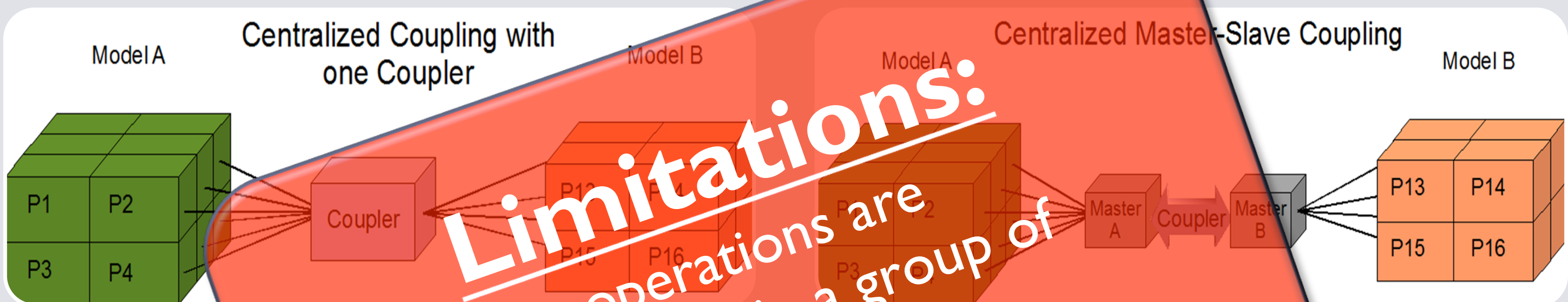
Type of Model Coupling

- **Communication Approach.**
  - Data exchange using message passing.
  - Suitable for HPC.
  - Libraries uses intrusive code.
  - Could use coupler or not.



Bettencourt (MCEL), 2002

Coupling in HPC



Centralized Coupler:

One or a group of processes act as the coupler.

Centralized Master-Slave Coupler:

Every slave process communicates through a master. A master acts as the coupler.

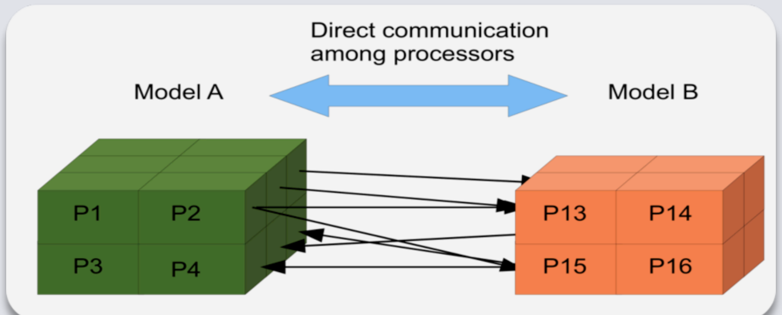
Limitations:

All coupling operations are concentrated only in a group of processes:

- Produce potentially bottlenecks.
- Underuse of computational resources.
- Memory bounds model resolution.

Coupling in HPC

Distributed Coupling



Distributed Coupling:

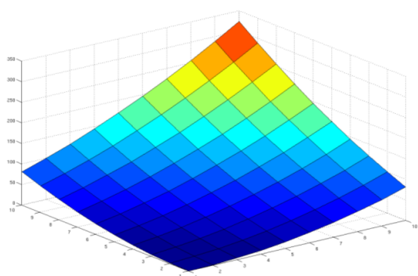
Every process communicates directly with processes into the another model.

- No dedicated or centralized Coupler.
- No synchronization.
- Direct communication leads to better scalability.
- Resolution is balanced. No bound by memory.
- Coupling operations are performed in parallel.

Profiling Performance

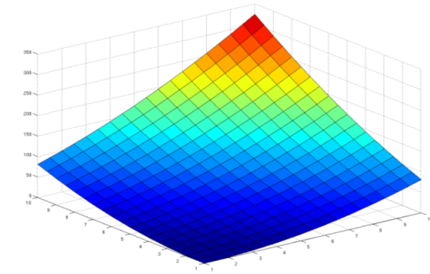
The bilinear interpolation was implemented and used to test and verify the characteristics mentioned.

Original Function
Coarse Mesh



Bilinear Interpolation

Interpolated Function
Fine Mesh



San Diego State University

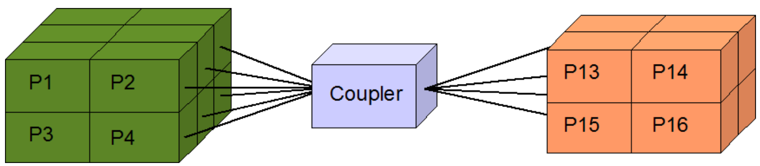
CSRC

Office of Science
U.S. Department of Energy

Berkeley Lab

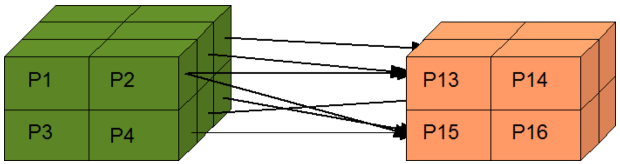
Profiling Performance

Centralized Approach (CA)



- **Model A:** Processes allocate memory for the subdomain coarse mesh, evaluate the initial function, and send the function info to the Coupler.
- **Coupler:** Allocates both the complete coarser and the fine meshes. Receives from the Model A processors, performs the interpolation and sends the result to their respective Model B processors.
- **Model B:** Processes allocate the subdomain fine mesh and receive the interpolated function values from the coupler.

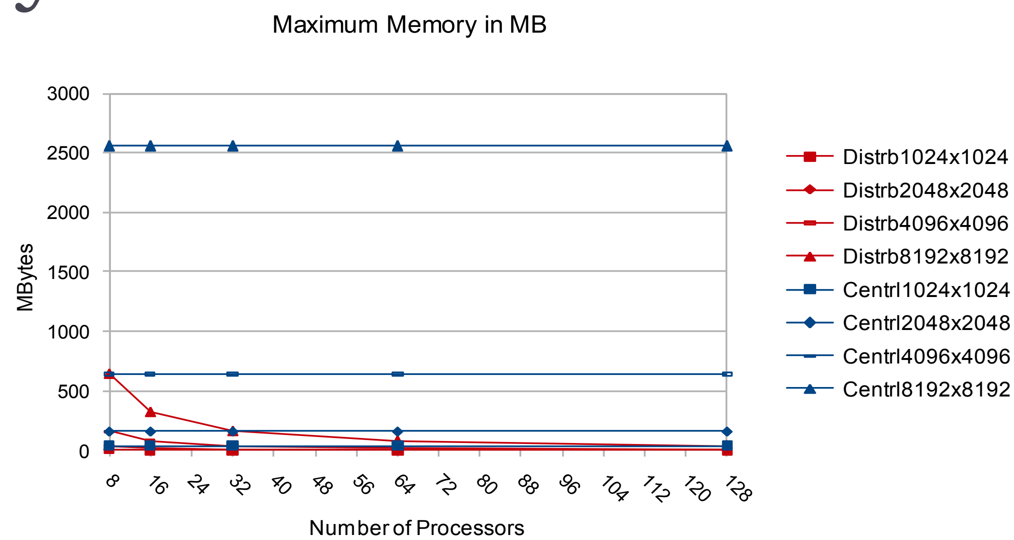
Distributed Approach (DA)

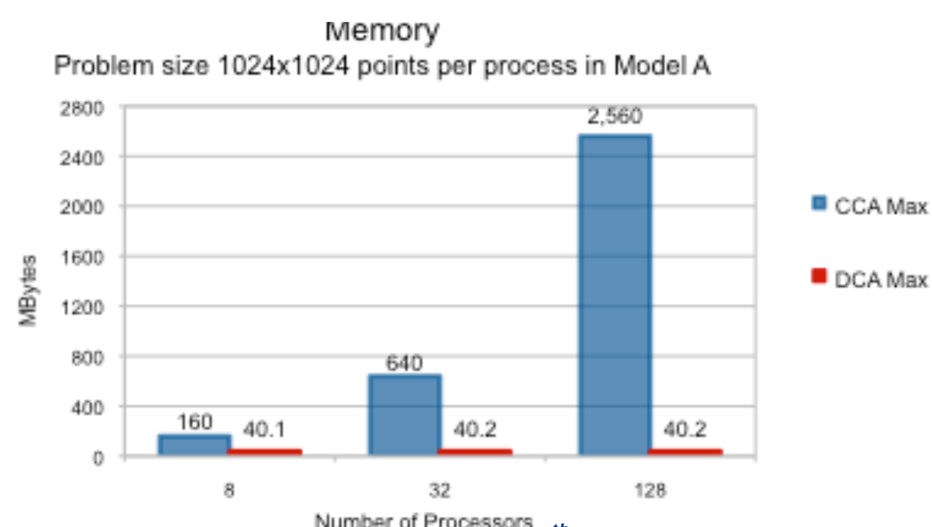


- **Model A:** Processes allocate memory for subdomain coarse mesh, evaluate the initial function, and send the function info to their respective Model B processes.
- **Model B**: Processors allocate memory for both the subdomain fine mesh and the coarse mesh corresponding to the area to be interpolated. They Receive from different senders the information about the function, and interpolate the function.

Profiling Memory use

- CA, large amount of memory concentrated in a single processor. A problem can run out of memory.
- Under DA, memory requirements are distributed among processors. Larger problems can be handled increasing number of processors.

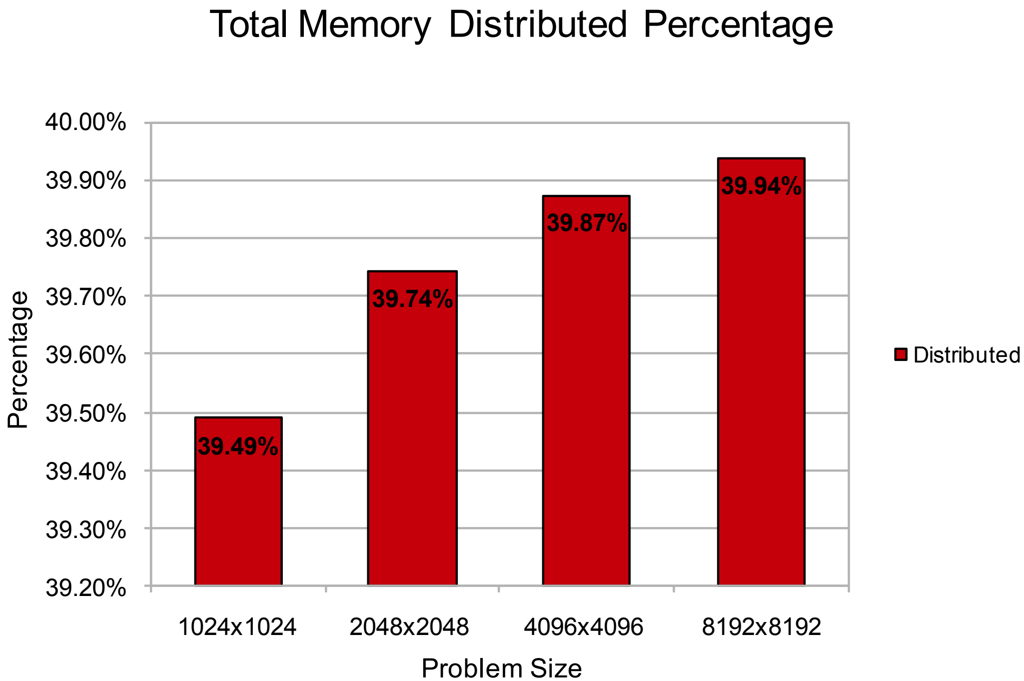




San Diego State University

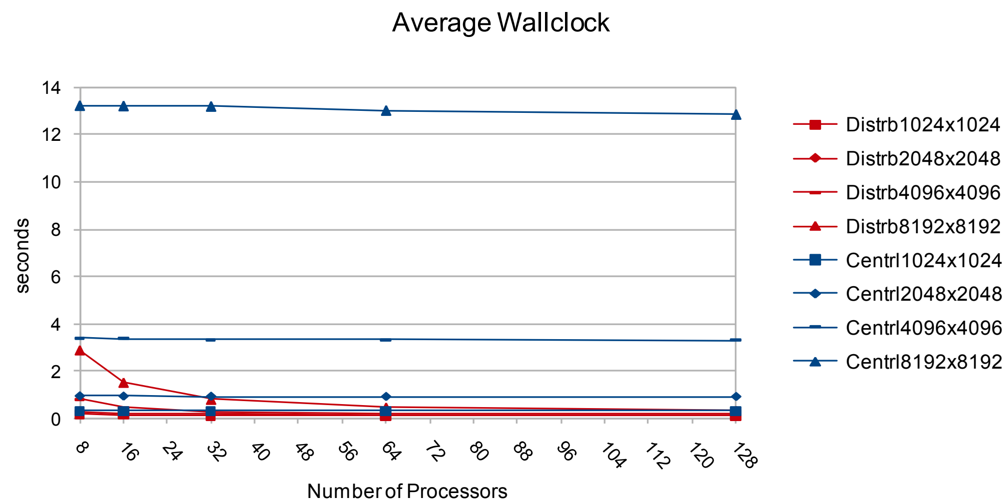
Profiling Memory use

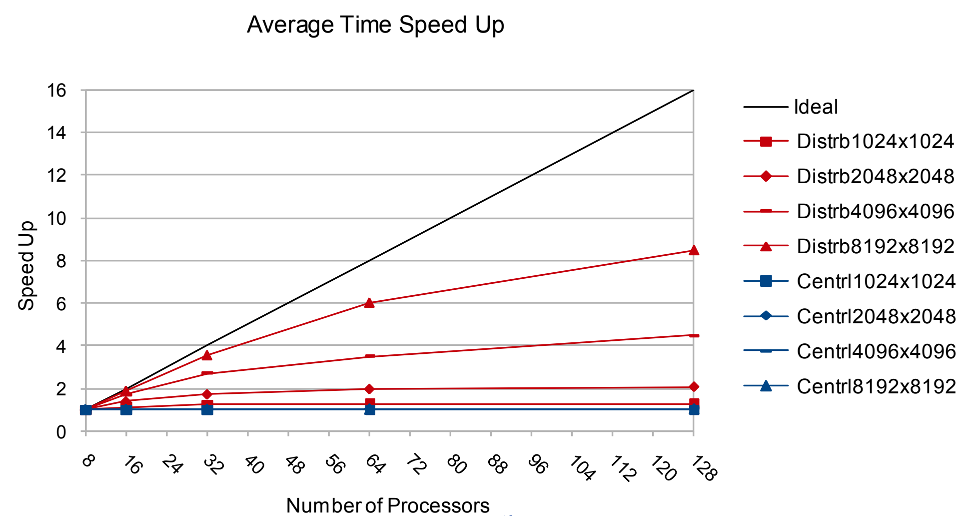
- In terms of total memory allocated, DA uses 40% less memory than CA.



Profiling Execution time and MFLOPS

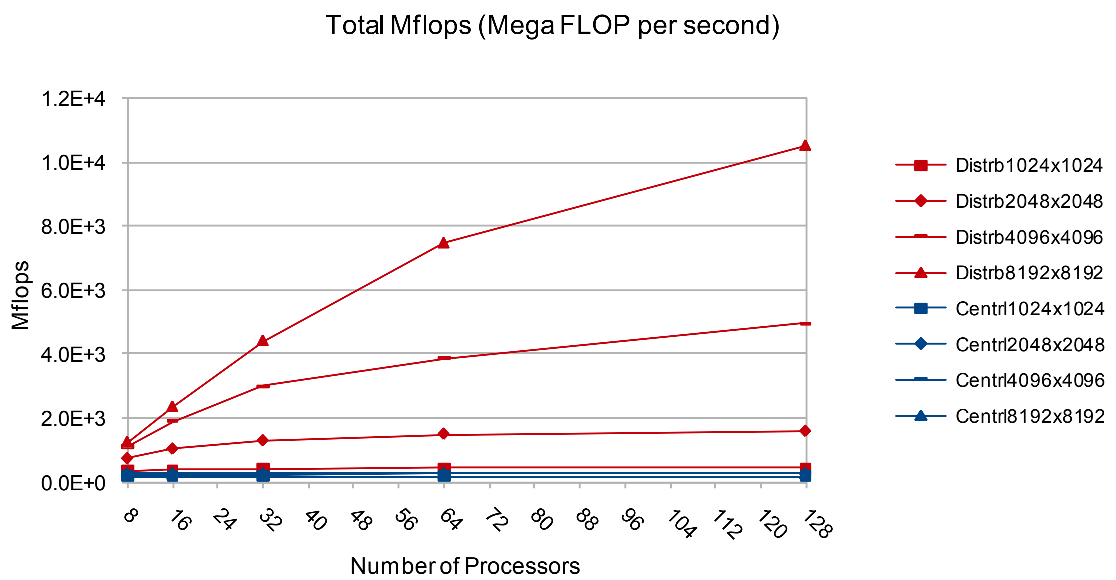
- CA does not reduce the processing time while DA does.
- DA gets speed up of total processing which is better when the problem size increases.





Experiments & Results: Performance

- Under DA The larger is the problem size the more increasing of MFLOPS.



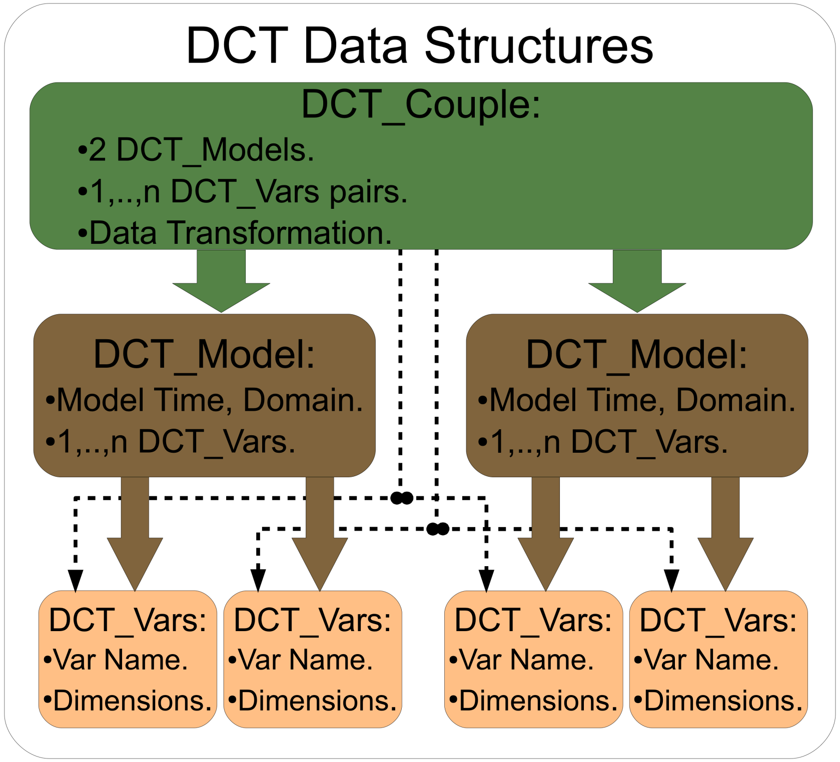
Distributed Coupling Toolkit (DCT)

- Library to Coupling Model.
- The Coupling is pure distributed.
- Non-intrusive lines code.
- Coupling numerical models:
  - ✓ With different resolutions in time and space.
  - ✓ That manage field and variables located differently in the grid cells.
  - ✓ Using different kinds of domain grids and variables discretized differently.
  - ✓ With different schedules to exchange information.
  - ✓ Parallelized independently using different data layout and distribution in arbitrary number of processors.

- A key contribution is the coupling operations are purely distributed
- Our approach comprises of TWO strategies:
  - ✓ Interrelated Data Structures.
  - ✓ Definition of different phases.

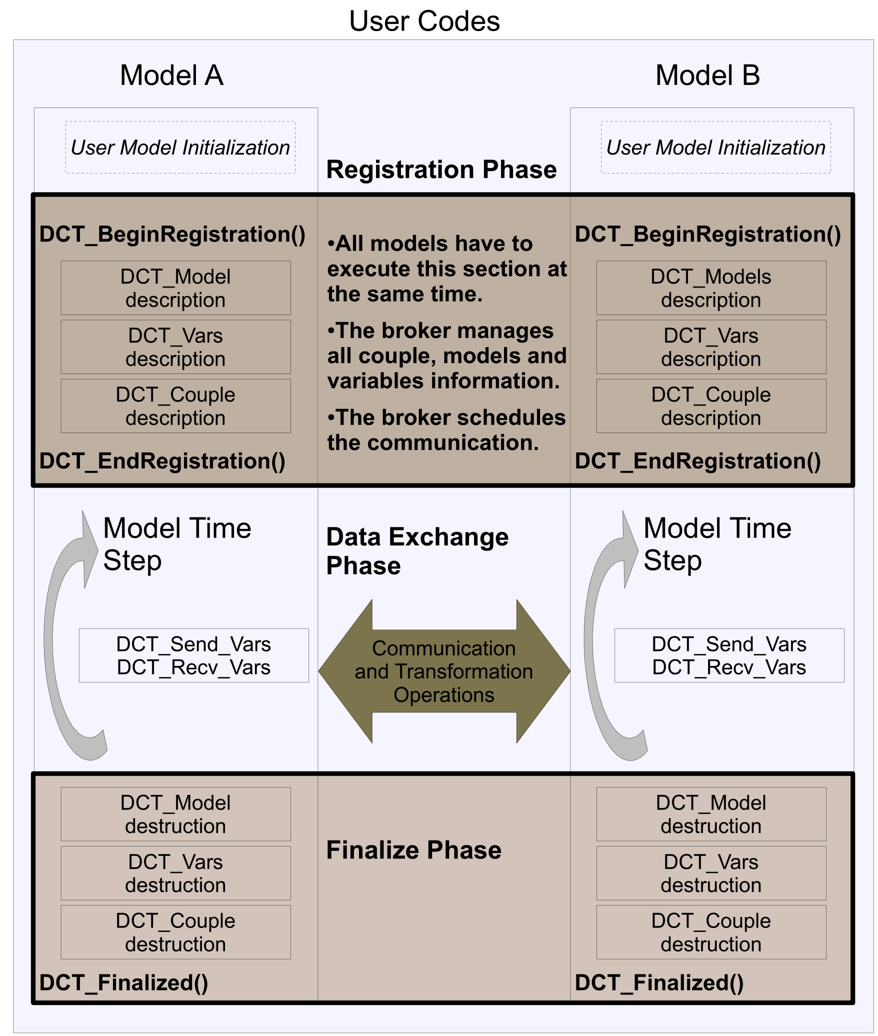
Distributed Coupling Toolkit (DCT)

- DCT Data Structures:
  - ✓ Capture the essentials.
  - ✓ Describe user models.
  - ✓ Describe the coupling mechanism.



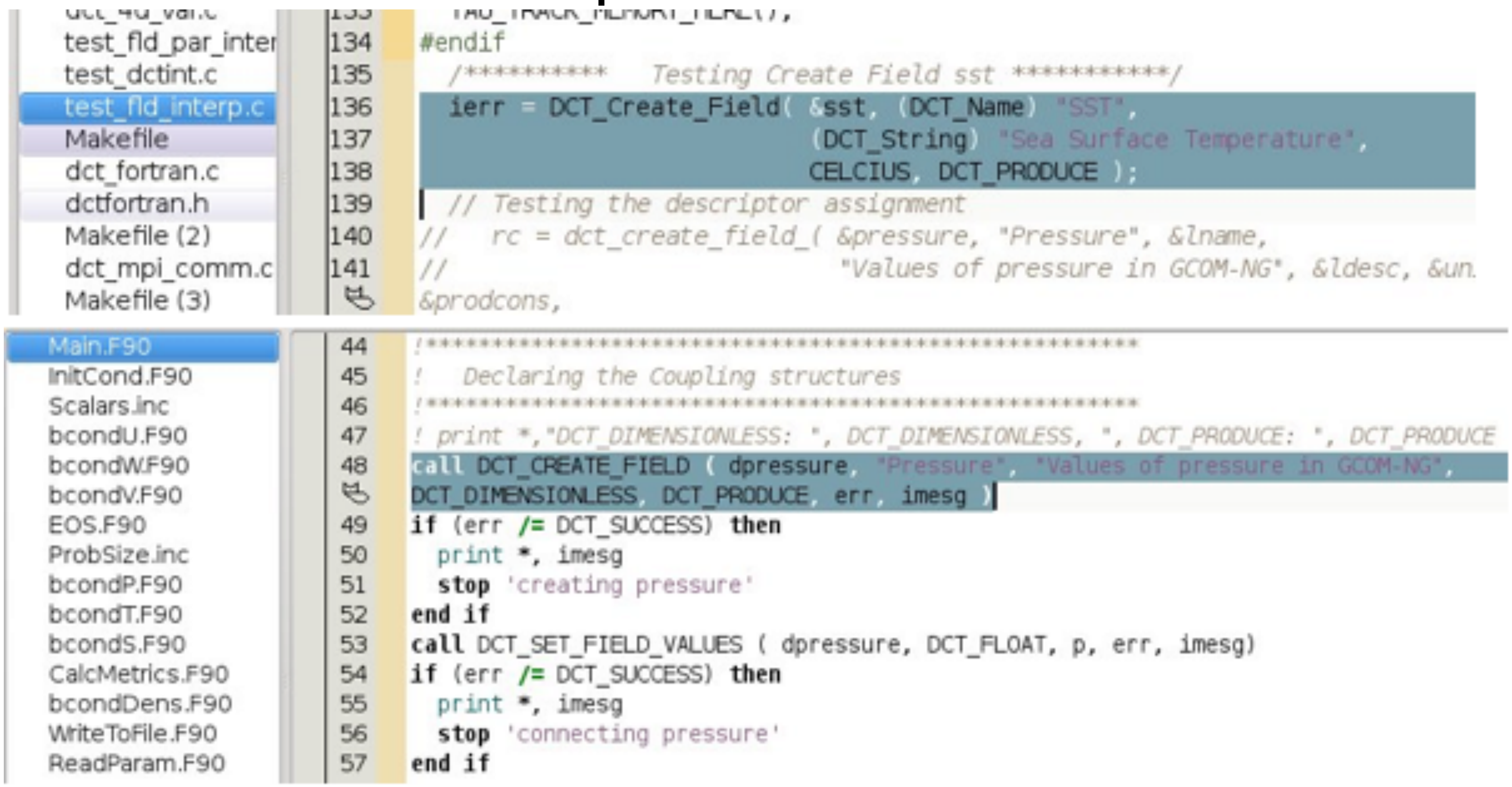
Distributed Coupling Toolkit (DCT)

- DCT Coupling Phases:
  - ✓ Separate the formulation from operations.
  - ✓ Organize the code.



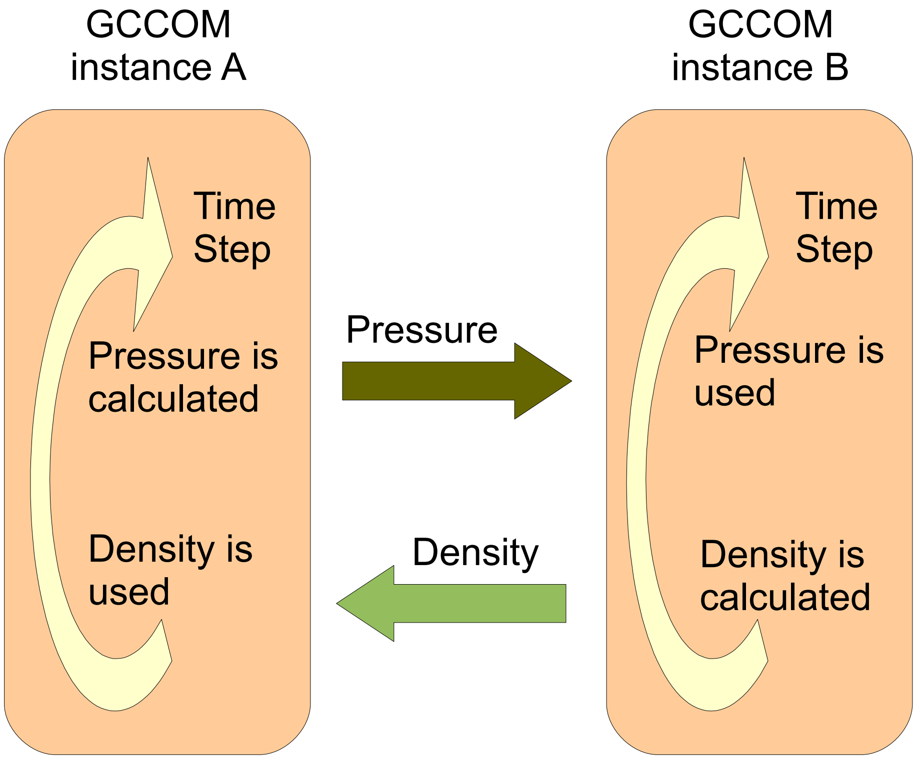
Distributed Coupling Toolkit (DCT)

- It is coded in ANSI C.
- A Fortran interface is provided.



Distributed Coupling Toolkit (DCT)

- Testing the library:
  - ✓ Two instances of GCCOM.
  - ✓ Running concurrently.
  - ✓ Results must be the same.



Distributed Coupling Toolkit (DCT)

Instance A — Variable declaration — **Instance B**

Variable declaration

Instance A:

```
TYPE (DCT_3D_VAR)    :: dpressure, ddensity
TYPE (DCT_MODEL)     :: dmodel
TYPE (DCT_COUPLE)    :: dcouple
DOUBLE PRECISION, DIMENSION (:,:,:) :: cplpres, cpldens
```

Instance B:

```
TYPE (DCT_3D_VAR)    :: dpressure, ddensity
TYPE (DCT_MODEL)     :: dmodel
TYPE (DCT_COUPLE)    :: dcouple
DOUBLE PRECISION, DIMENSION (:,:,:) :: cplpres, cpldens
```

DCT Registration Phase

Instance A:

```
call dct_beginregistration( DCT_FALSE, world_comm )

User creates DCT_Model dmodel, named “gccom_a”

User creates DCT_3d_Var dpressure, named
“Pressure_a”, to be produced

!!! ** Linking pressure to the model  **
call dct_set_model_var ( dmodel, dpressure, &
                         & DCT_3D_VAR_TYPE )

User creates DCT_3d_Var ddensity, named
“Density_a”, to be consumed

!!! ** Linking density to the model  **
call dct_set_model_var ( dmodel, ddensity, &
                         & DCT_3D_VAR_TYPE )

User creates DCT_Couple dcouple, named
“DenTemTest1”, linking dmodel and remote model
named “gcom_b”

!!! ***** Linking pressure to the couple  *****
call dct_set_coupling_vars ( dcouple, dpressure, &
              & "Pressure_b", DCT_3D_VAR_TYPE, &
              & DCT_NO_INTERPOLATION )

!!! ***** Linking density to the couple  *****
call dct_set_coupling_vars ( dcouple, ddensity, &
              & "Density_b", DCT_3D_VAR_TYPE, &
              & DCT_NO_INTERPOLATION )

call dct_endregistration ( )
```

Instance B:

```
call dct_beginregistration( DCT_TRUE, world_comm )

User creates DCT_Model dmodel, named “gccom_b”

User creates DCT_3d_Var dpressure, named
“Pressure_b”, to be consumed

!!! ** Linking pressure to the model  **
call dct_set_model_var ( dmodel, dpressure, &
                         & DCT_3D_VAR_TYPE )

User creates DCT_3d_Var ddensity, named
“Density_b”, to be produced

!!! ** Linking density to the model  **
call dct_set_model_var ( dmodel, ddensity, &
                        & DCT_3D_VAR_TYPE )

User creates DCT_Couple dcouple, named
“DenTemTest1”, linking dmodel and remote model
named “gcom_a”

!!! ***** Linking pressure to the couple  *****
call dct_set_coupling_vars ( dcouple, dpressure, &
              & "Pressure_a", DCT_3D_VAR_TYPE, &
              & DCT_NO_INTERPOLATION )

!!! ***** Linking density to the couple  *****
call dct_set_coupling_vars ( dcouple, ddensity, &
              & "Density_a", DCT_3D_VAR_TYPE, &
              & DCT_NO_INTERPOLATION )

call dct_endregistration ( )
```

DCT Data Exchange Phase

Instance A:

```
call dct_update_model_time( dmodel )

!!! Producing Pressure
cplpres = p( 1:vnpts(1), 1:vnpts(2), 1:vnpts(3) )
call dct_send_3d_var ( dpressure )

!!! Consuming Density
call dct_recv_3d_var ( ddensity )
dens( 1:vnpts(1), 1:vnpts(2), 1:vnpts(3) ) = cpldens
```

Instance B:

```
call dct_update_model_time( dmodel )

!!! Consuming Pressure
call dct_recv_3d_var ( dpressure )
p( 1:vnpts(1), 1:vnpts(2), 1:vnpts(3) ) = cplpres

!!! Producing Density
cpldens = dens( 1:vnpts(1), 1:vnpts(2), 1:vnpts(3) )
call dct_send_3d_var ( ddensity )
```

DCT Finalize Phase

Instance A:

```
User destroys all created variables; dcouple,dmodel,
dpressure, ddensity

call dct_finalized ( )
```

Instance B:

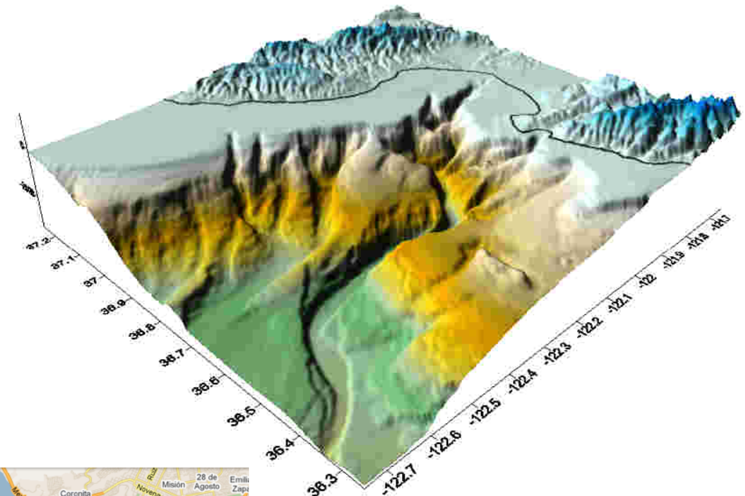
```
User destroys all created variables; dcouple,dmodel,
dpressure, ddensity

call dct_finalized ( )
```

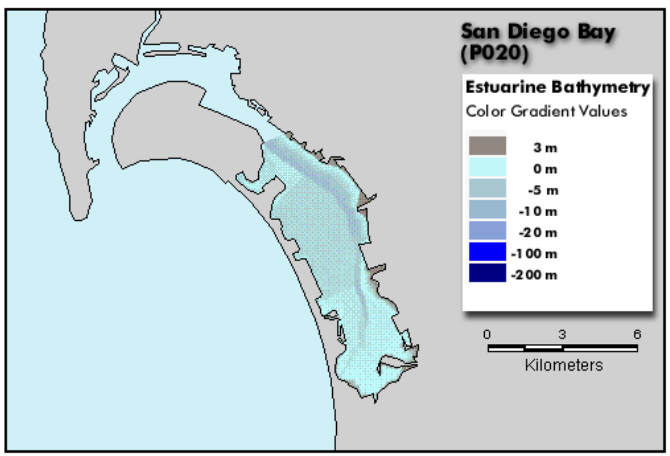
Objectives:

- ✓ Test the DCT phase effectiveness.
- ✓ Test the Fortran interface.
- ✓ Test the communication functions.

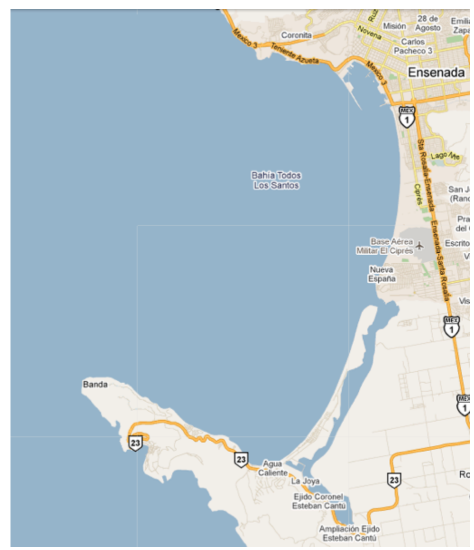
Applications

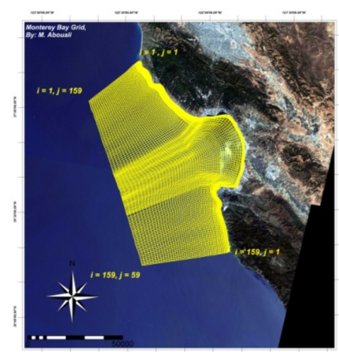
- Suggested Cases Studies:
  - ✓ Monterrey Bay.
  - ✓ San Diego Bay.
  - ✓ Todos los Santos Bay.





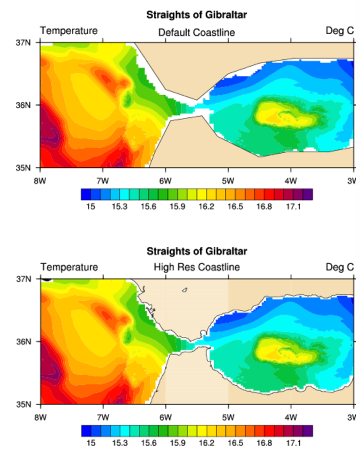
Courtesy NOAA



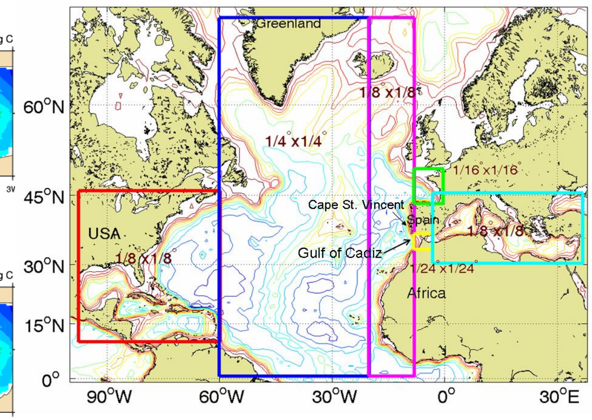


Applications

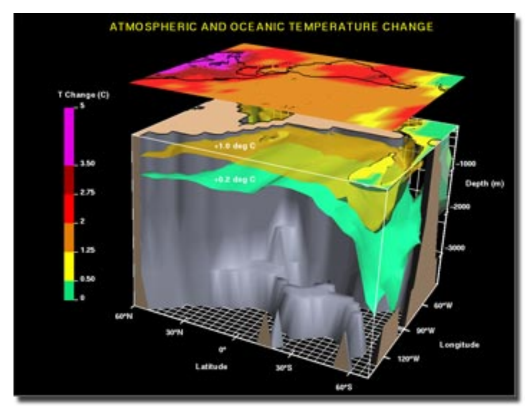
- Types of Couplings:
  - ✓ Circulation Variability: Multi-resolution Coupling: DieCAST-GCCOM, POM-GCCOM.
  - ✓ Combined effect atmosphere-ocean: Multi-physics coupling: GCAM-GCCOM.



Courtesy NCAR



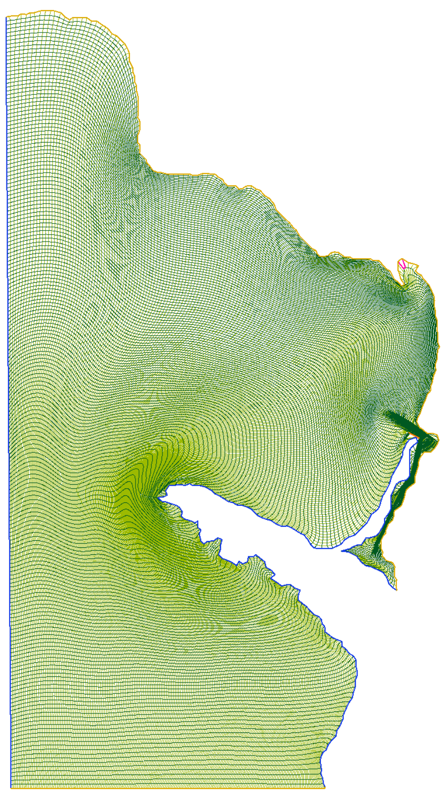
Courtesy MEDiNA web page



Courtesy NOAA

Applications

- Multi-resolution coupling:
  - Todos los Santos Bay.

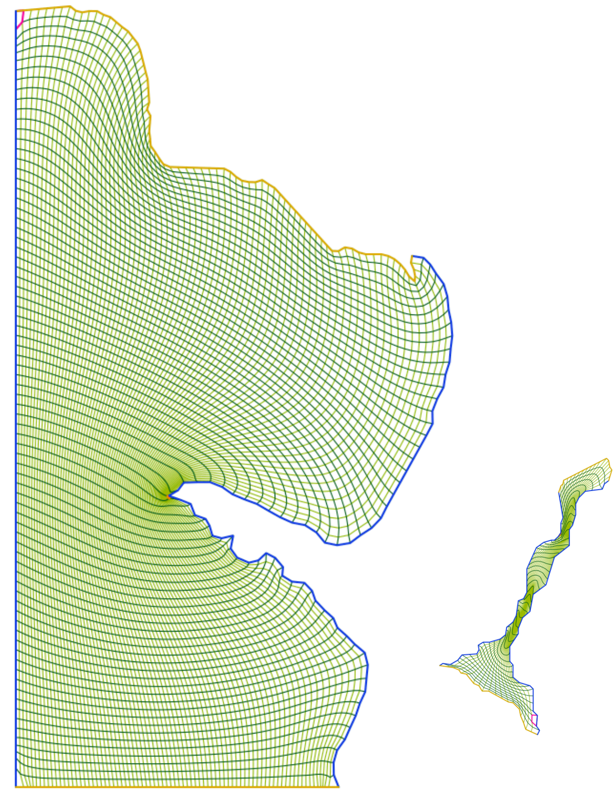


Applications

- Multi-resolution coupling:
  - Todos los Santos Bay.

Applications

- Multi-resolution coupling:
  - Todos los Santos Bay.



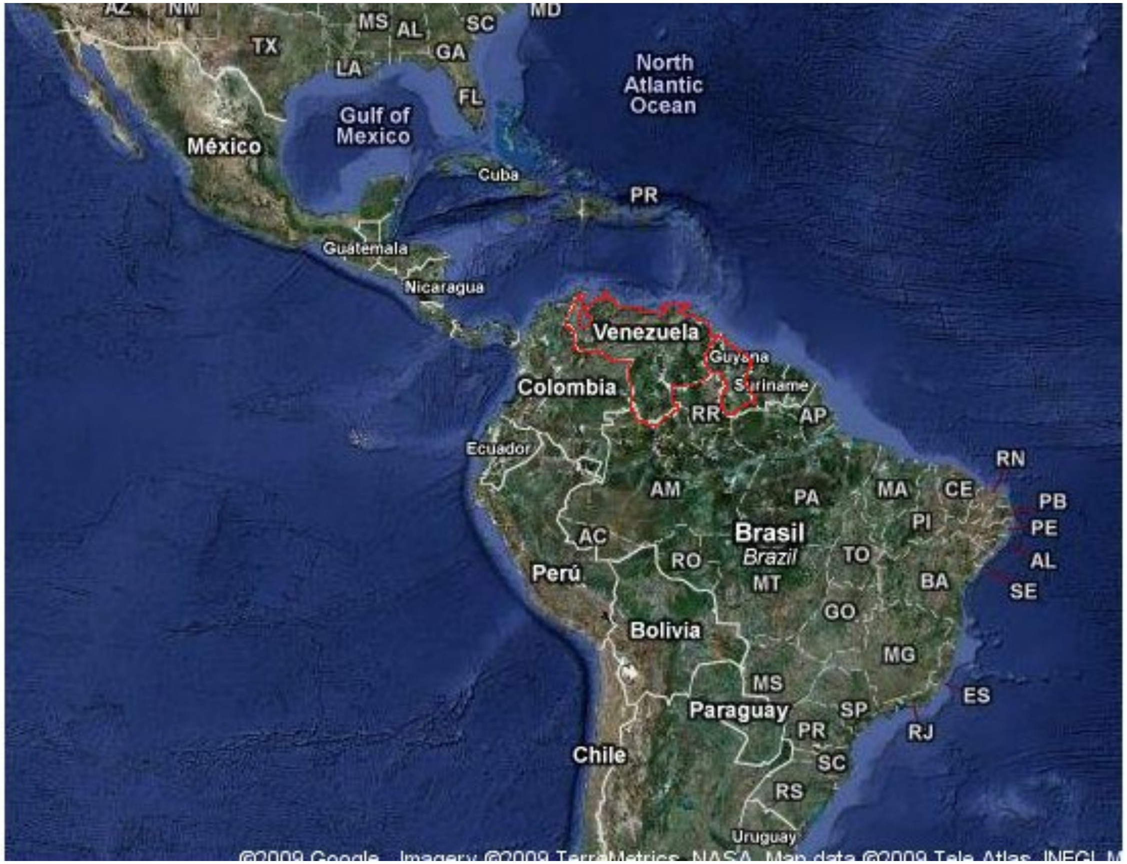
Summary and Future Plans

- GCEM is a ongoing project to model coastal environments.
- GCEM provides high resolution and boundary fitting ocean and atmosphere models.
- The weak model coupling was introduced.
- For HPC, we choose the pure distributed approach.
- DCA overcomes the drawbacks showed by CCA.
- DCT is a library to couple models under HPC environments.

Summary and Future Plans

- Two key strategies are used to address the desired features of DCT.
- Some functionalities and the interface was tested using an earlier version of GCCOM.
- Some cases of study with different kind of coupling was suggested.
- More functionalities are planned to be implemented:
  - Support for wider type of meshes.
  - Different operations; such that regriding mapping, interpolation, filters, etc.

Thank You!!!!!


North Atlantic Ocean
Gulf of Mexico
México
TX
LA
MS
AL
GA
SC
FL
MD
AZ
NM
Cuba
PR
Guatemala
Nicaragua
Venezuela
Guyana
Suriname
Colombia
Ecuador
Perú
Bolivia
Brasil
Brazil
Paraguay
Chile
Uruguay
RR
AP
AM
PA
MA
CE
RN
PB
PE
AL
SE
PI
AC
RO
MT
TO
BA
GO
MG
MS
ES
SP
RJ
PR
SC
RS

Thank You!!!!!

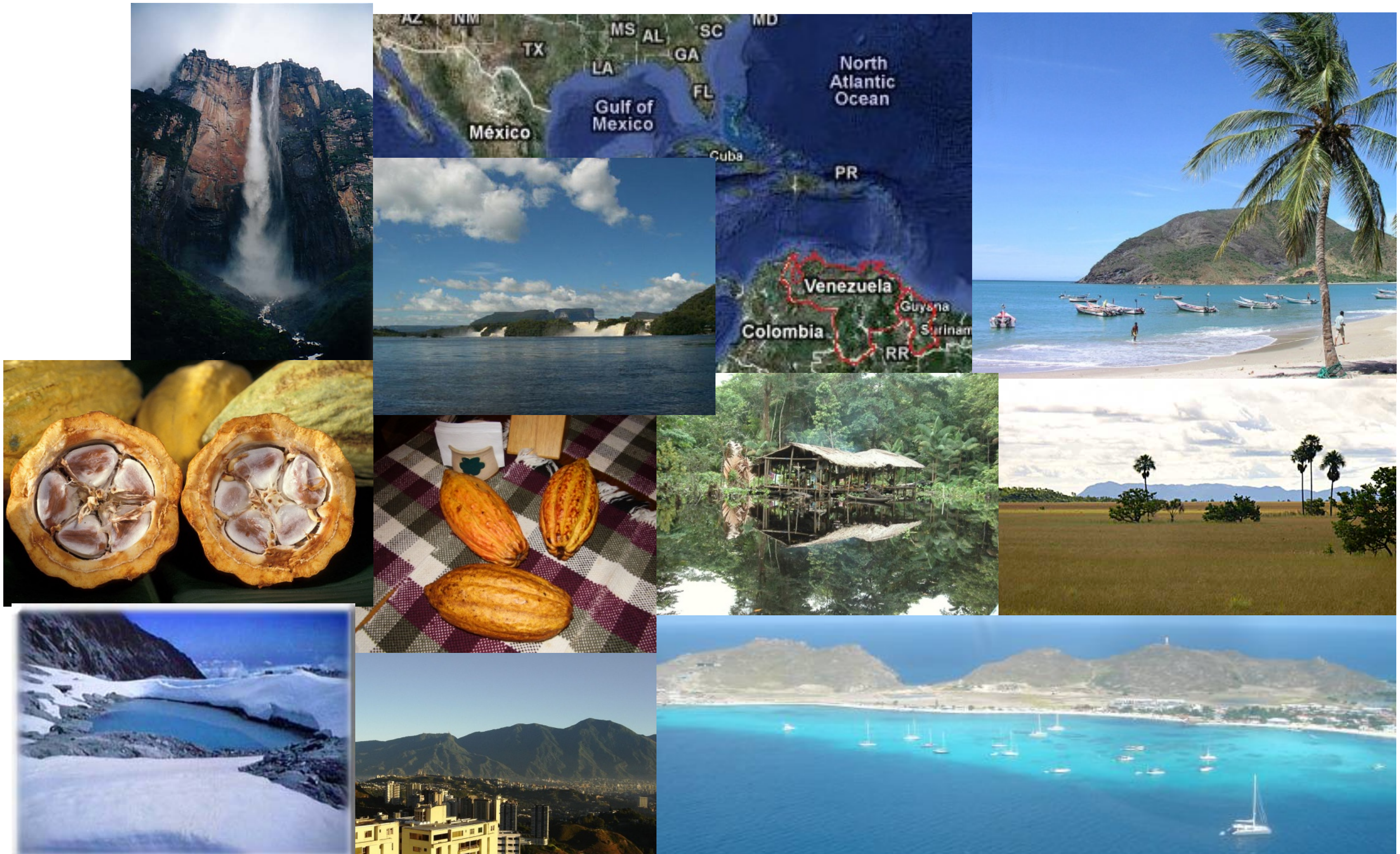
San Diego State University

CSRC

Office of Science
U.S. Department of Energy

Berkeley Lab

Thank You!!!!!



San Diego State University

an interdisciplinary gateway to the future CSRC

Office of Science U.S. Department of Energy

Berkeley Lab

Acknowledgment

Special Thanks to:

- Prof. José Castillo.
- Dr. Tony Drummond.
- Team mates.
- P244 and Lunch Time mates.

How many times must a man look up
Before he can see the sky?
Yes, 'n' how many ears must one man have
Before he can hear people cry?
Yes, 'n' how many deaths will it take till he knows
That too many people have died?
The answer, my friend, is blowin' in the wind
The answer is blowin' in the wind…

Bob Dylan, Blowin' in the wind. (1962)

San Diego State University
CSRC
Office of Science
U.S. Department of Energy
Berkeley Lab