Components: Coupling and Concurrency in ESMF

Nancy Collins
nancy@ucar.edu
CAS2K5
14 Sept 2005

http://www.esmf.ucar.edu
Outline

- Brief ESMF Overview and Status
- ESMF Components
- Future Directions
What is ESMF?

- ESMF provides tools for turning model codes into **components** with standard interfaces and standard drivers.
- ESMF provides data structures and **common utilities** that components use for routine services such as data communications, regridding, time management and message logging.

**ESMF GOALS**

1. Increase scientific productivity by making model components much easier to build, combine, and exchange, and by enabling modelers to take full advantage of high-end computers.
2. Promote new scientific opportunities and services through community building and increased interoperability of codes (impacts in collaboration, code validation and tuning, teaching, migration from research to operations)
Growing ESMF Customer Base

- Original ESMF applications:
  - NOAA GFDL atmospheres
  - NOAA GFDL MOM4 ocean
  - NOAA NCEP atmospheres, analyses
  - NASA GMAO models and GEOS-5
  - NASA/COLA Poseidon ocean
  - LANL POP ocean
  - NCAR WRF
  - NCAR CCSM
  - MITgcm atmosphere and ocean

- Other groups using ESMF:
  - NASA GISS
  - UCLA
  - CSU
  - NASA Land Information Systems (LIS) project
  - NOAA Integrated Dynamics in Earth’s Atmosphere (IDEA) project, more…

- New applications coming in during FY05 through the newly funded, ESMF-based DoD Battlespace Environments Institute (BEI):
  - DoD Navy HYCOM ocean
  - DoD Navy NOGAPS atmosphere
  - DoD Navy COAMPS coupled atm-ocean
  - DoD Air Force GAIM ionosphere
  - DoD Air Force HAF solar wind
  - DoD Army ERDC WASH123 watershed

- More new applications will begin adopting ESMF during FY06 through the ESMF-based NASA Modeling Analysis and Prediction (MAP) Climate Variability and Change program.

- Further growth of the customer base is anticipated through development of an ESMF-based Space Weather computational environment.
ESMF Development Status

- Overall architecture is well-defined and well-accepted
- Components and low-level communications stable
- Concurrency and multiple executables supported
- Logically rectangular grids with regular and arbitrary distributions implemented
- On-line parallel regridding (bilinear, 1\textsuperscript{st} order conservative) completed and optimized
- Other parallel methods, e.g. halo, redistribution, low-level comms implemented
- Utilities such as time manager, logging, and configuration manager usable and adding features
- Virtual machine with interface to shared / distributed memory implemented, hooks for load balancing implemented
ESMF Near-Term Priorities, FY05/06

• Reworked design and implementation of array / grid / field interfaces and array-level communications
• Optimized regridding and low-level communications
• Support for representing, partitioning, communicating with, and regridding unstructured grids and semi-structured (tripole, cubed-sphere) grids
• Read/write interpolation weights and grid specifications
Planned ESMF Extensions

- Looser couplings: extended support for multiple executables and Grid-enabled versions of ESMF
- Support for advanced I/O, including support for asynchronous I/O, checkpoint/restart, and multiple archival mechanisms (e.g. NetCDF, HDF5, binary, etc.)
- Advanced support for data assimilation systems, including data structures for observational data and adjoints for ESMF methods
- Support for nested, moving grids and adaptive grids
- Support for regridding in three dimensions and between different coordinate systems
- Advanced optimization and load balancing
ESMF and Prism, CCA

- **PRISM** is a European Earth system modeling infrastructure project
- Complementary effort to ESMF
- Centralized “Coupler” All-to-All
- More structure for ensuring user data is interoperable
- ESMF/PRISM collaboration on CF conventions and interoperable components

- **CCA** is a U.S. Department of Energy SciDAC project
- Interfaces and tools to generate dynamic general scientific components
- Quick to assemble new applications
- Collaborators include LANL, ANL, LLNL, ORNL, Sandia, University of Tennessee, and many more.
- ESMF components running as CCA Components demonstrated at SC03. Ongoing ESMF collaboration with CCA on F90/C++ interoperability.

For joint use with PRISM, ESMF developed a component database to store component import/export fields and component descriptions
Computational Characteristics of Weather/Climate

- Mix of global transforms and local communications
- Load balancing for diurnal cycle, event (e.g. storm) tracking
- Applications typically require 10s of GFLOPS, 100s of PEs – but can go to 10s of TFLOPS, 1000s of PEs
- Required Unix/Linux platforms span laptop to Earth Simulator
- Multi-component applications: component hierarchies, ensembles, and exchanges; components in multiple contexts
- Data and grid transformations between components
- Applications may be MPMD/SPMD, concurrent/sequential, combinations
- Parallelization via MPI, OpenMP, shmem, combinations
- Large applications (typically 100,000+ lines of source code)
What is a Component in ESMF?

- User code with a 3 defined subroutines
  - Initialize()
  - Run()
  - Finalize()

- Takes data from an Import State, produces data in an Export State
  - Does not directly exchange data otherwise
Components as Building Blocks

- Hierarchical Applications
  - Compose nested/larger applications more easily
- Modularity
  - No internal dependencies on other components
- Coupling/Concurrency
  - Users write coupler code, decide on the control flow
- Isolation
  - Components pick their own internal communication strategy regardless of how others do it
- Uniform Communication API
  - VM hides dependencies on shared memory, distributed memory, clusters, single processor machines
Application Example: GEOS-5 AGCM

- Each box is an ESMF component
- Every component has a standard interface so that it is swappable
- Data in and out of components are packaged as state types with user-defined fields
- New components can easily be added to the hierarchical system
- Coupling tools include regridding and redistribution methods
Components Address Parallelism

• Component writers face a daunting set of parallelism choices
  – Serial vs. parallel execution
  – Sequential vs. concurrent
  – Single vs. multiple executable

• Component code remains unchanged
  – Different drivers and couplers are needed but do not affect the heart of the computational code
Sequential vs Concurrent
Creating Ensembles

Multiple Components running the same module code. Each can compute with a different set of input parameters.
UCLA AGCM - MIT OGCM

December–February SST forecasts initialized with early June oceanic conditions

Contour interval: 1° C; +/- 0.5° C contours are also shown.
UCLA AGCM - LANL POP

December–February SST forecasts with POP (early June oceanic initial conditions)

SST anomaly forecast, 1998

Observed SST anomaly, 1998

SST anomaly forecast, 1999

Observed SST anomaly, 1999

SST anomaly forecast, 2000

Observed SST anomaly, 2000

SST anomaly forecast, 2001

Observed SST anomaly, 2001

SST anomaly forecast, 2002

Observed SST anomaly, 2002
Infrastructure Timeline: 
ESMF Path

- **ESMF common modeling infrastructure (~5 year)**
  Technical foundation that allows for organized and exchangeable codes
- **Set of modeling codes that are based on ESMF (~5 year)**
  Community pool of interoperable science components with which to assemble applications
- **Multi-agency and international organization (~5 year)**
  Organizational foundation on which to establish collaborations and set priorities
- **Standards for model and data description (~10 year)**
  Prerequisite for an advanced modeling and collaboration environment that includes knowledge management
- **ESMF part of an established end-to-end community-based modeling environment (~20 year)**
  An Earth System Knowledge Environment (ESKE) that combines models, data, experiments, collaborative tools and information resources in a way that fosters knowledge sharing and accelerates scientific workflow
For More Information

Main project web site:
http://www.esmf.ucar.edu

For scheduling and release information, see:
http://www.esmf.ucar.edu > Development

This includes latest releases, known bugs, supported platforms.

Task lists, bug reports, and support requests are tracked on the ESMF SourceForge site:
http://sourceforge.net/projects/esmf
## ESMF Framework Layers

**Applications:** unified collection of Components

<table>
<thead>
<tr>
<th>Layer</th>
<th>Data classes</th>
<th>Operations</th>
<th>Communication</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component Layer</td>
<td>Components</td>
<td>Component Methods</td>
<td>Component Communication</td>
<td>Collective I/O</td>
</tr>
<tr>
<td>Supports large-scale task parallelism and interoperability of frameworks: INTEROPERABILITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library Layer</td>
<td>Parallel data types</td>
<td>Parallel methods</td>
<td>Library communication</td>
<td>I/O Library Interface</td>
</tr>
<tr>
<td>Parallel constructs which hide the details of data distribution and movement: EASE OF USE</td>
<td>(plus Grid, Map, Index, and Iterator classes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Utility Layer</td>
<td>Local Data Types</td>
<td>Serial methods</td>
<td>Utility communication</td>
<td>I/O Utilities</td>
</tr>
<tr>
<td>Basic building blocks which are highly optimized and may contain machine dependencies: PORTABILITY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Machine Model:** abstraction of physical computer
Field test: M.I.T. General Circulation Model (MITgcm) to NCAR Community Atmospheric Model (CAM).

Versions of CAM and MITgcm were adapted to
a. have `init()`, `run()`, `finalize()` interfaces
b. accept, encode and decode `ESMF_state` variables

A coupler component that maps MITgcm grid to CAM grid was written

Runtime steps
1. MITgcm prepares export state.
2. Export state passes through parent to coupler.
3. Coupler returns CAM gridded SST array which is passed as import state to CAM gridded component.

- Uses `ESMF_GridComp`, `ESMF_CplComp`, and `ESMF_Regrid` sets of functions.