



A Software Developing Environment for Earth System Modeling

Depei Qian
Beihang University
CScADS Workshop, Snowbird, Utah
June 27, 2012



Outline



Motivation

Purpose and Significance

Research Contents

Technology Roadmap

IDE design & development functions

ESM Module Encapsulation

Runtime Support

Deployment

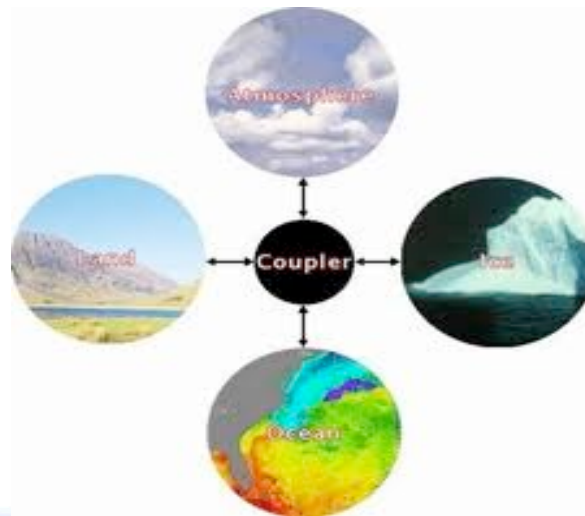




Motivation

➤ CCSM/CESM

- ◆ Coupled climate models used to simulate the Earth's climate system
- ◆ Includes four separate models: the atmosphere, oceans, land surface and sea ice
- ◆ Support the research of global climate change in the past, present and future





Motivation (cont'd)

Lack of integrated HPC software development environment for earth system modeling

Primitive program development approaches

Various model development tools

Program development coupled with machine hardware / software



Graphical IDE is in urgent need

1. All-in-one platform, comprehensive tool library and the module / template library, to improve development efficiency
2. Decoupled with software/hardware details to allow researchers to focus on model development
3. Customized development environment to provide lifecycle support in developing the earth system models





Outline



Motivation

Purpose and Significance

Research Contents

Technology Roadmap

IDE design & development functions

ESM Module Encapsulation

Runtime Support

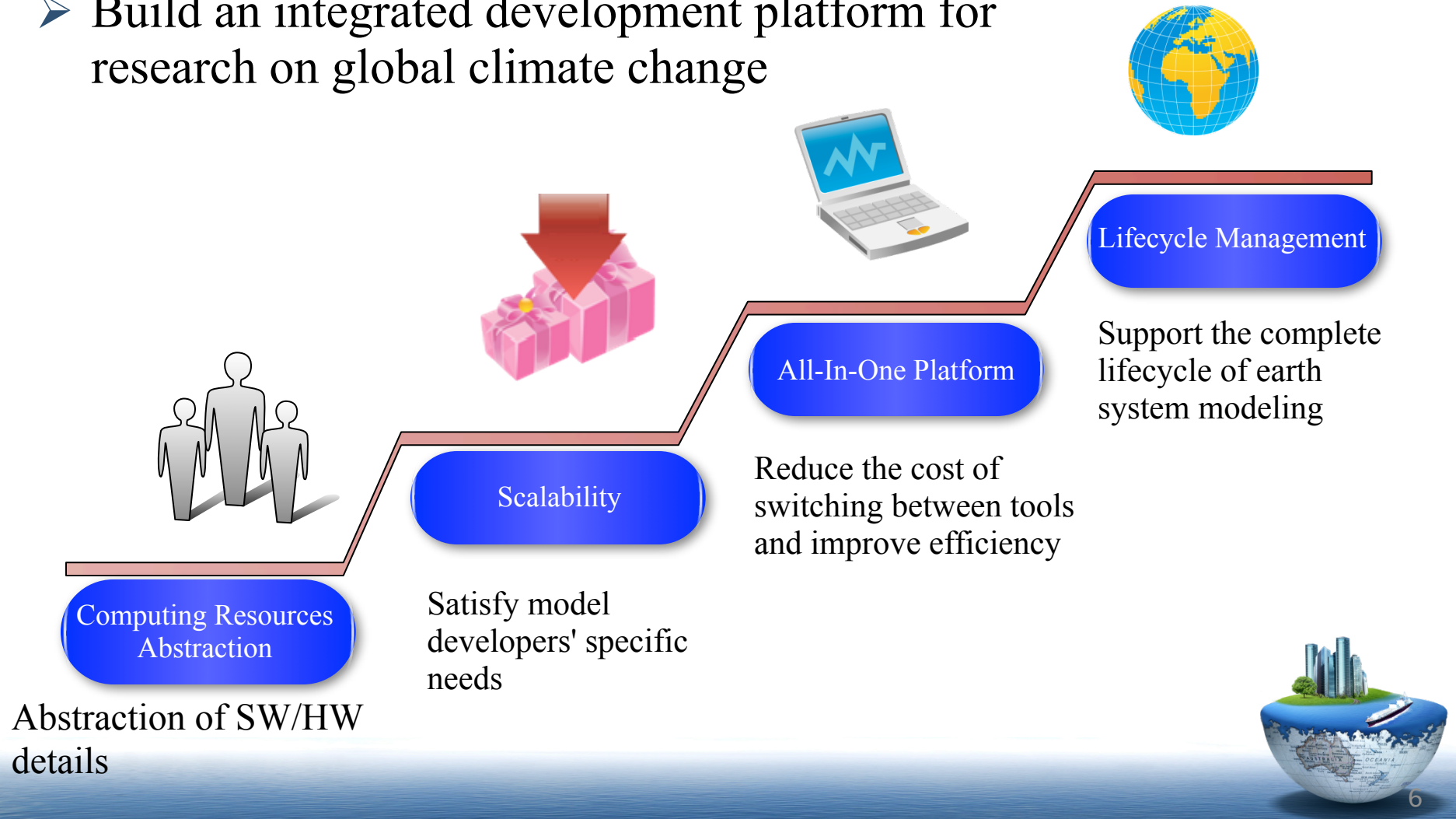
Deployment





Purpose and significance

- Focus on support to the development of earth system models
- Build an integrated development platform for research on global climate change





Outline



Motivation

Purpose and Significance

Research Contents

Technology Roadmap

IDE design & development functions

ESM Module Encapsulation

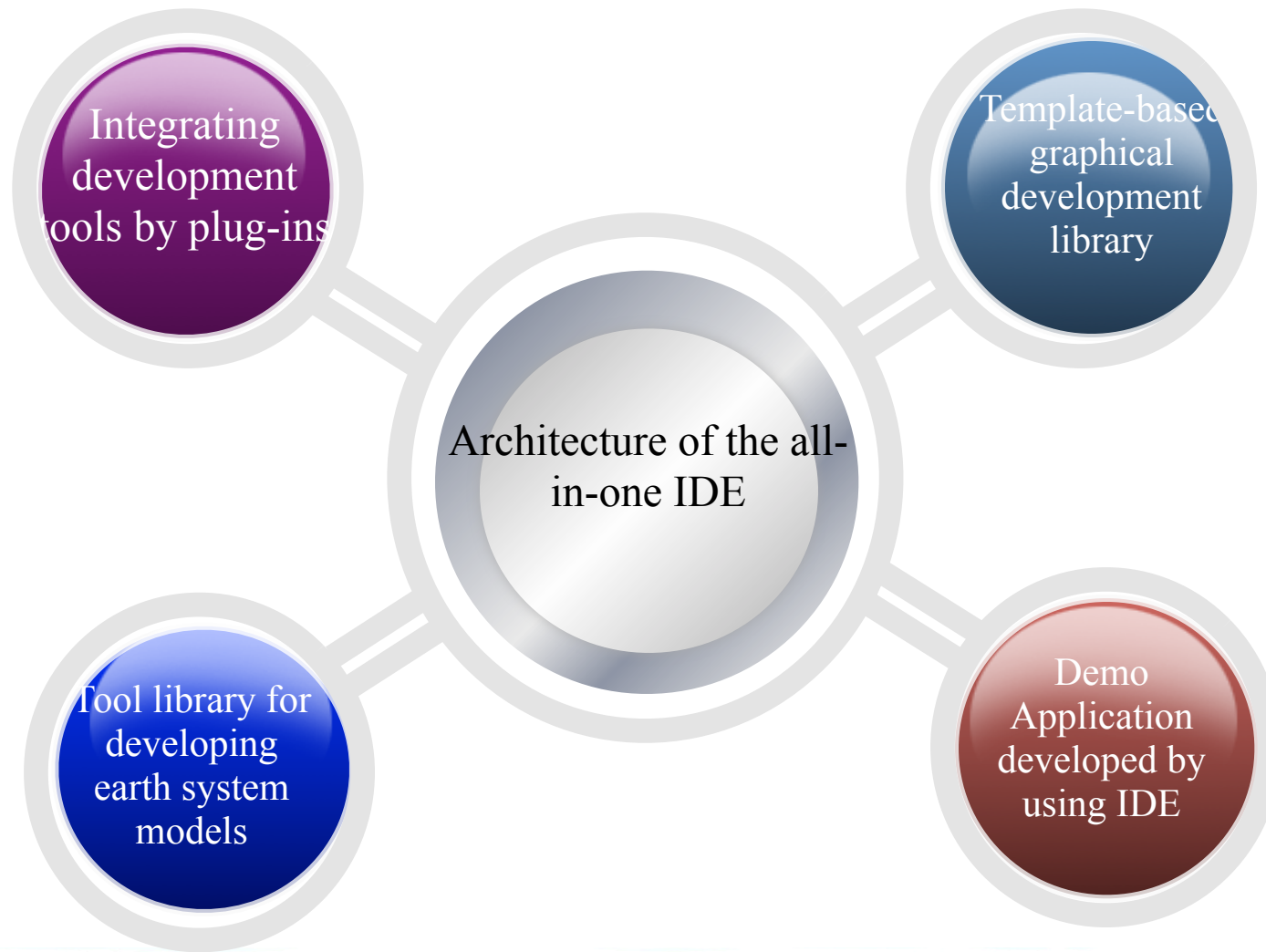
Runtime Support

Deployment





Research content





Outline



Motivation

Purpose and Significance

Research Contents

Technology Roadmap

IDE design & development functions

ESM Module Encapsulation

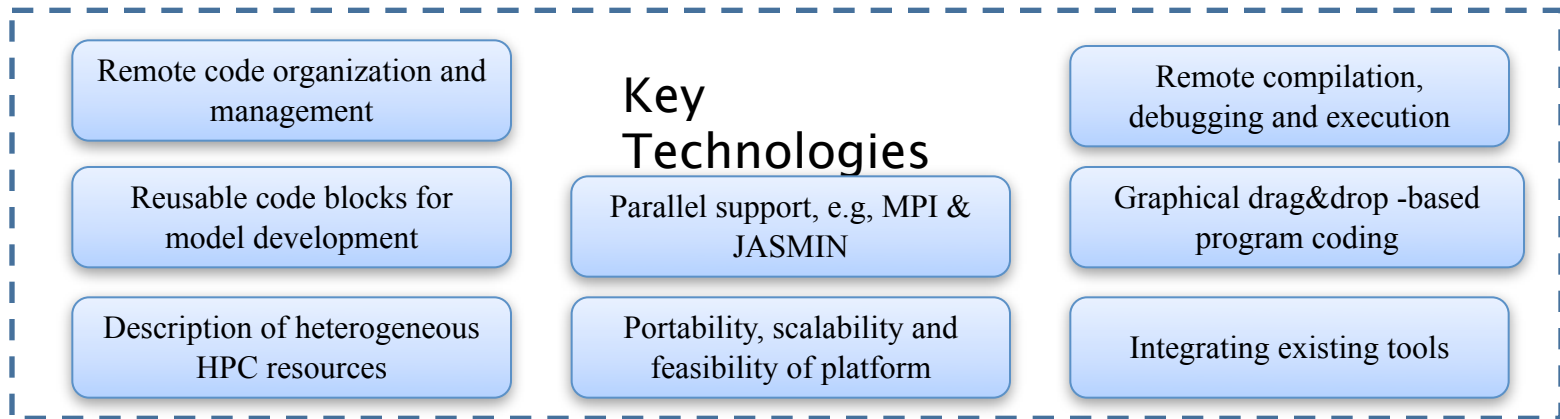
Runtime Support

Deployment





Technology roadmap

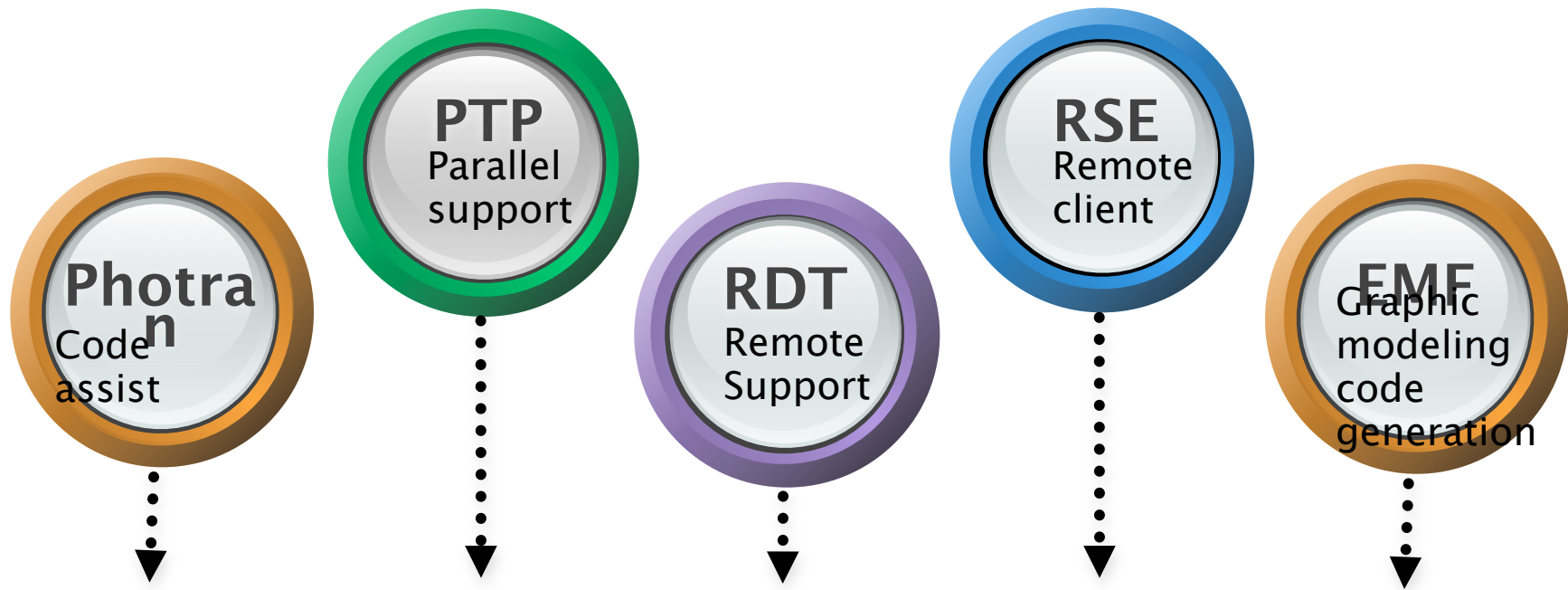


Plug-in Integration & Graphical interface support





Technology roadmap



Overall strategy

Based on the open source Eclipse: mature community, abundant references, plenty of free plug-ins, wide range of languages





Outline





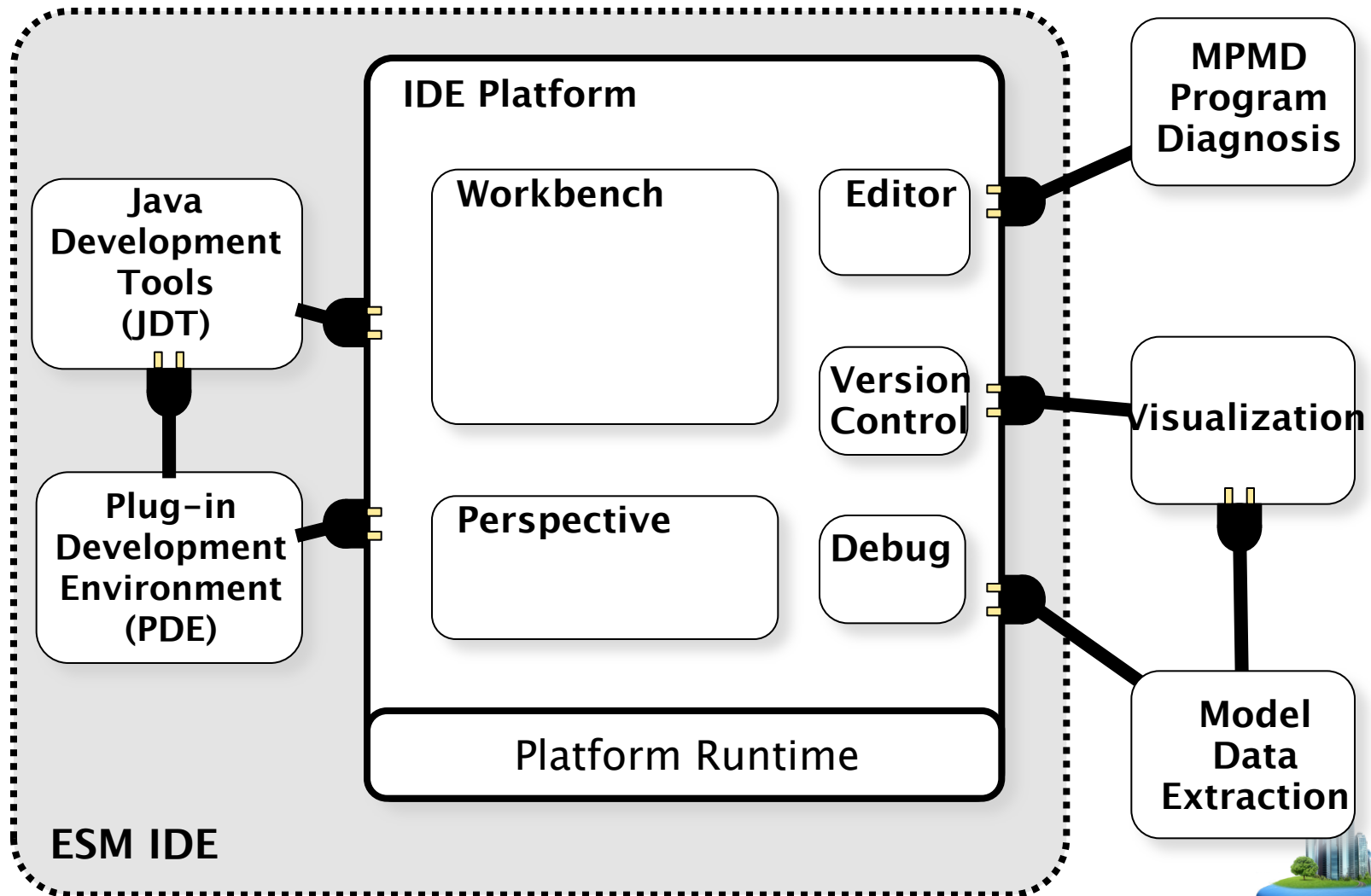
IDE design & development functions

- Works have been done :
 - ◆ IDE overall design
 - ◆ Fortran code assistant
 - ◆ Source code version control
 - ◆ Remote online multi-collaboration
 - ◆ Integration of other useful tools





IDE overall design





Fortran code assistant

ESM Development - pop/source/diagnostics.F90 - Eclipse Platform

File Edit Refactor Navigate Search Project Run Window Help

ESM Perspective

Project Explorer

- BNU_ESM.new
 - remote_BNU_ESM.new
 - source_BNU_ESM [INTEGRATED] BNU_ESM.new
 - models
 - atm
 - bld
 - cpl
 - cpl6
 - inputdata.F90 13 11-4-2 T
 - BitCheck_mod.F90 13 11-4-2 T
 - diag_mod.F90 13 11-4-2 T
 - doc
 - doc.api

VI Edit

Highlighted & File Layout

diagnostics.F90

```
use timers
use exit_mod

implicit none
private
save

!!PUBLIC MEMBER FUNCTIONS:

public :: init_diagnostics,
diag_init_sums,
diag_global_preupdate
diag_global_afterupdate
diag_print,
diag_transport,
cfl_advect,
cfl_vdiff,
cfl_hdiff,
cfl_check,
check_KE

!!PUBLIC DATA MEMBERS:
```

baroclinic.F90

```
use advection, only: advu, advt
use pressure_grad, only: lpressu
use horizontal_mix, only: hdiff
use vertical_mix, only: vmix_coef
use vdiff, only: vdiff, impvmixt, impmixt
use umix_kpp, only: add_kpp_source
use diagnostics, only: ldiag_cfl
DIAG_KE_ADV_2D, DIAG_KE_PRES
DIAG_TRACER_HDIFF_2D, DIAG_KE_PRES
DIAG_TRACER_SFC_FLX, DIAG_KE_PRES

use state_mod, only: state
use ice, only: liceform, ice_forcing
use time_management, only: mix_ratio
gamma, c2dtt

use io_types, only: nml_in, nml_out
use tavg, only: define_tavg_fields
use forcing, only: STF, SMF, lstrat
use forcing_shf, only: SHF_QSW, SHF_QSW
! use forcing_coupled, only: forcing_coupled
use forcing_pt_interior, only: forcing_pt_interior
use forcing_s_interior, only: forcing_s_interior
use exit_mod, only: sigAbort, sigError
```

Outline

- diagnostics
 - transport
 - init_diagnostics
 - diag_init_sums
 - diag_global_preupdate
 - diag_global_afterupdate
 - diag_print
 - diag_transport
 - cfl_advect

History

/pop in svn://124.205.18.230/jsi

修订	日期	作者	注释
*63	11-5-12 上午9...	ys	New imj

Console

```
wangln@210.31.66.230
drwx----- 5 wangln Model 4096 May 2 06:46 doc
drwx----- 7 wangln Model 4096 May 2 06:46 inputdata
drwx----- 7 wangln Model 4096 May 2 06:46 inputdata_user
drwx----- 11 wangln Model 4096 May 2 06:45 models
-rw----- 1 wangln Model 1362 May 2 06:46 README
drwx----- 4 wangln wangln 4096 May 4 11:43 scripts
-rw----- 1 wangln Model 3145 May 2 06:46 SVN_EXTERNAL_DIRECTORIES
[wangln@login02 BNU_ESM]$
```

File: C:\setup.log

Command mode: _____

10:55
2011/5/13



Source code version control

The screenshot displays the Eclipse IDE interface for a project named "ESM Development - pop/source/diagnostics.F90". The main workspace shows two source files: "diagnostics.F90" and "baroclinic.F90". The "diagnostics.F90" file is open in the editor, showing Fortran code with comments and function definitions. The "baroclinic.F90" file is also open, showing similar code. The Project Explorer on the left shows the project structure, including subdirectories like "remote_BNU_ESM.new", "source_BNU_ESM", "models", "atm", "bld", "cpl", and "cpl6". The Remote Systems view shows a connection to a remote system "wangln@210.31.66.230". The History view is highlighted with a red dashed box, showing a table of revisions. The Console view at the bottom shows the output of a command, listing the contents of the "pop" directory.

Version Control & Change History

修订	日期	作者	注释
*63	11-5-12 上午9...	ys	New im

```
drwx----- 5 wangln Model      4096 May  2 06:46 doc
drwx----- 7 wangln Model      4096 May  2 06:46 inputdata
drwx----- 7 wangln Model      4096 May  2 06:46 inputdata_user
drwx----- 11 wangln Model     4096 May  2 06:45 models
-rw-----  1 wangln Model      1362 May  2 06:46 README
drwx----- 4 wangln wangln     4096 May  4 11:43 scripts
-rw-----  1 wangln Model      3145 May  2 06:46 SVN_EXTERNAL_DIRECTORIES
[wangln@login02 BNU_ESM]$
```




Integrated tools

ESM Development - pop/source/diagnostics.F90 - Eclipse Platform

File Edit Refactor Navigate Search Project Run Window Help

Project Explorer

- BNU_ESM.new
 - remote_BNU_ESM.new
 - source_BNU_ESM [Model/BNU_ESM]
 - models
 - atm
 - bld
 - cpl
 - cpl6
 - diag_mod.F90 13 11-4-2
 - data_mod.F90 13 11-4-2
 - bitCheck_mod.F90 13 11-4
 - diag_mod.F90 13 11-4-2
 - doc
 - doc.api

diagnostics.F90

```
use timers
use exit_mod

implicit none
private
save

!!PUBLIC MEMBER FUNCTIONS:

public :: init_diagnostics,
diag_init_sums,
diag_global_preupdate
diag_global_afterupdate
diag_print,
diag_transport,
cfl_advect,
cfl_vdiff,
cfl_hdiff,
cfl_check,
check_KE

!!PUBLIC DATA MEMBERS:
```

baroclinic.F90

```
use advection, only: advu, advt
use pressure_grad, only: lpress
use horizontal_mix, only: hdiff
use trimesh_x11

use
use
```

Outline

- diagnostics
- transport

Chesapeake Bay Bathymetry meters

Console

```
wangln@210.31.66.230
drwx----- 5 wangln Model 4096 May 2 06:46 doc
drwx----- 7 wangln Model 4096 May 2 06:46 inp
drwx----- 7 wangln Model 4096 May 2 06:46 inp
drwx----- 11 wangln Model 4096 May 2 06:45 mod
-rw----- 1 wangln Model 1362 May 2 06:46 REP
drwx----- 4 wangln wangln 4096 May 4 11:43 scr
-rw----- 1 wangln Model 3145 May 2 06:46 SV
[wangln@login02 BNU_ESM]$
```

File: C:\setup.log

Command mode: _____

10:55 2011/5/13

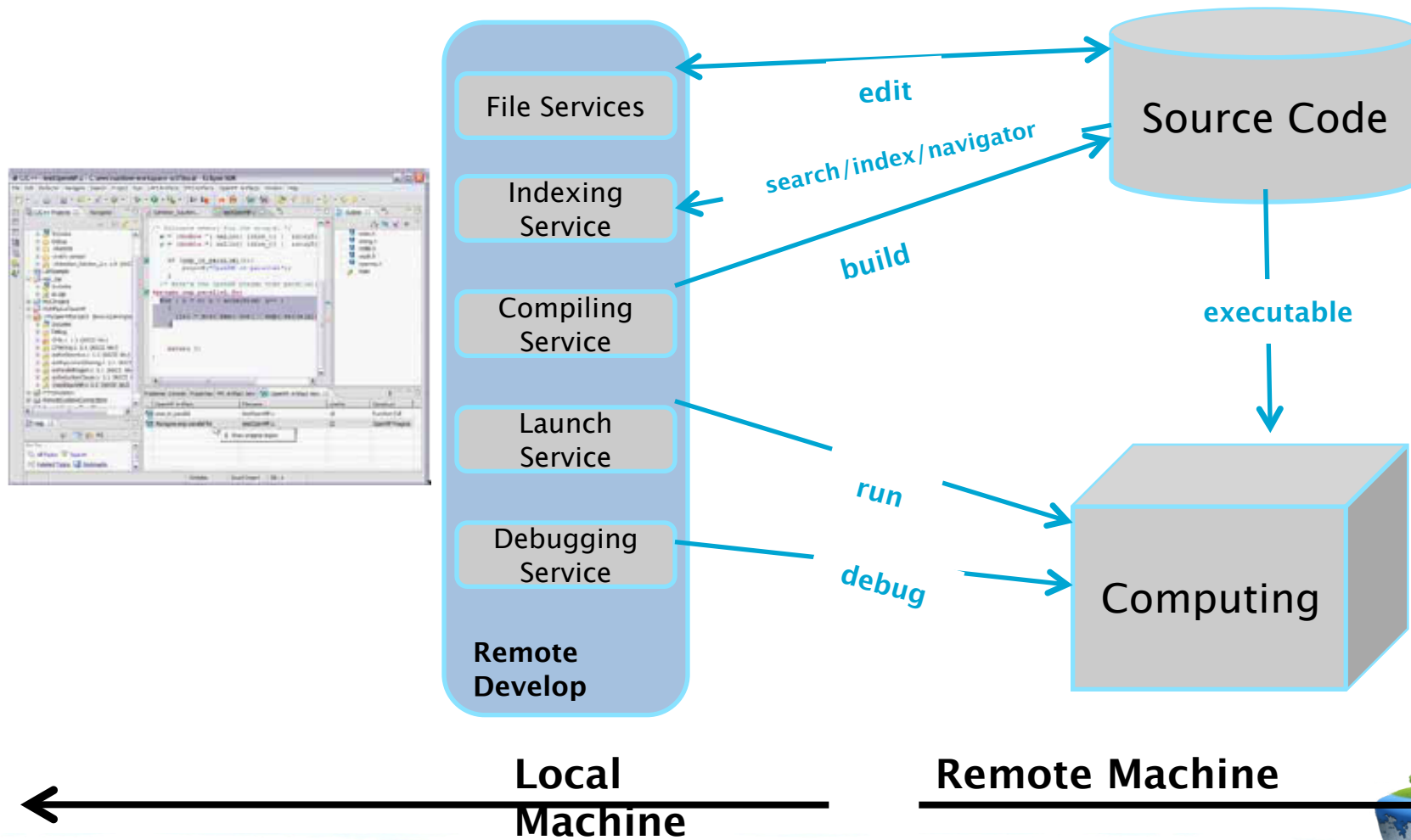
FTP Tools

Terminal



Remote function invocation

- Remote editing, compilation, debugging and execution





Outline



- Motivation
- Purpose and Significance
- Research Contents
- Technology Roadmap
- IDE design & development functions
- ESM Module Encapsulation**
- Runtime Support
- Deployment





ESM module encapsulation

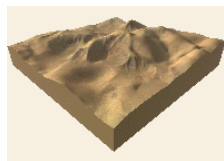
- Encapsulation of reusable code blocks for model development



Radiation Module



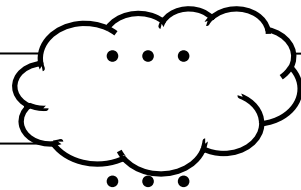
Time integration module



Boundary layer module



Coupler module



Solver module

Modular Encapsulation Specification



module unit

- High-performance
- Reusable



model module library

- Enrichment
- Optimization





Program decomposition

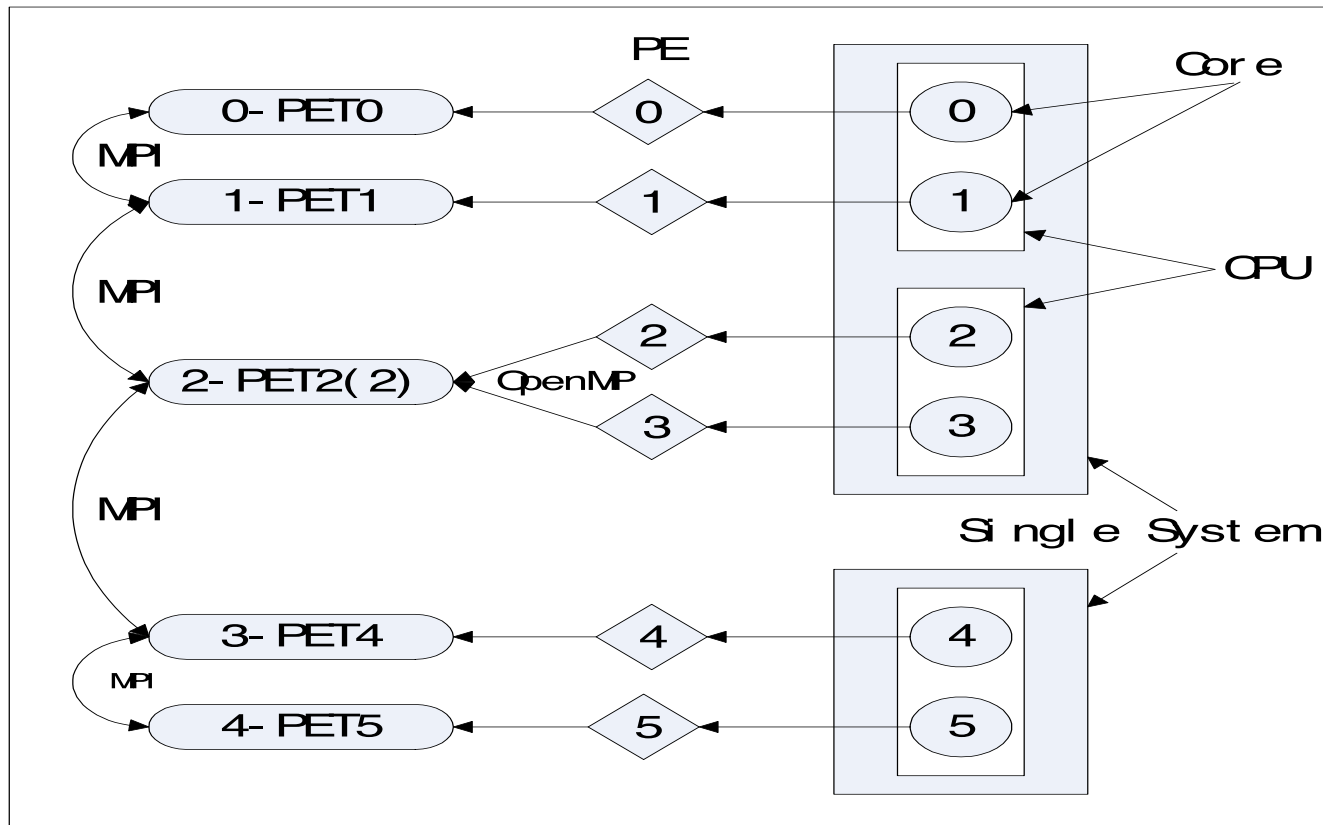
- Decomposition of POP (Parallel Ocean Program) into modules





Resource allocation

- Parallel computing resources allocation for POP components





Dynamic process organization

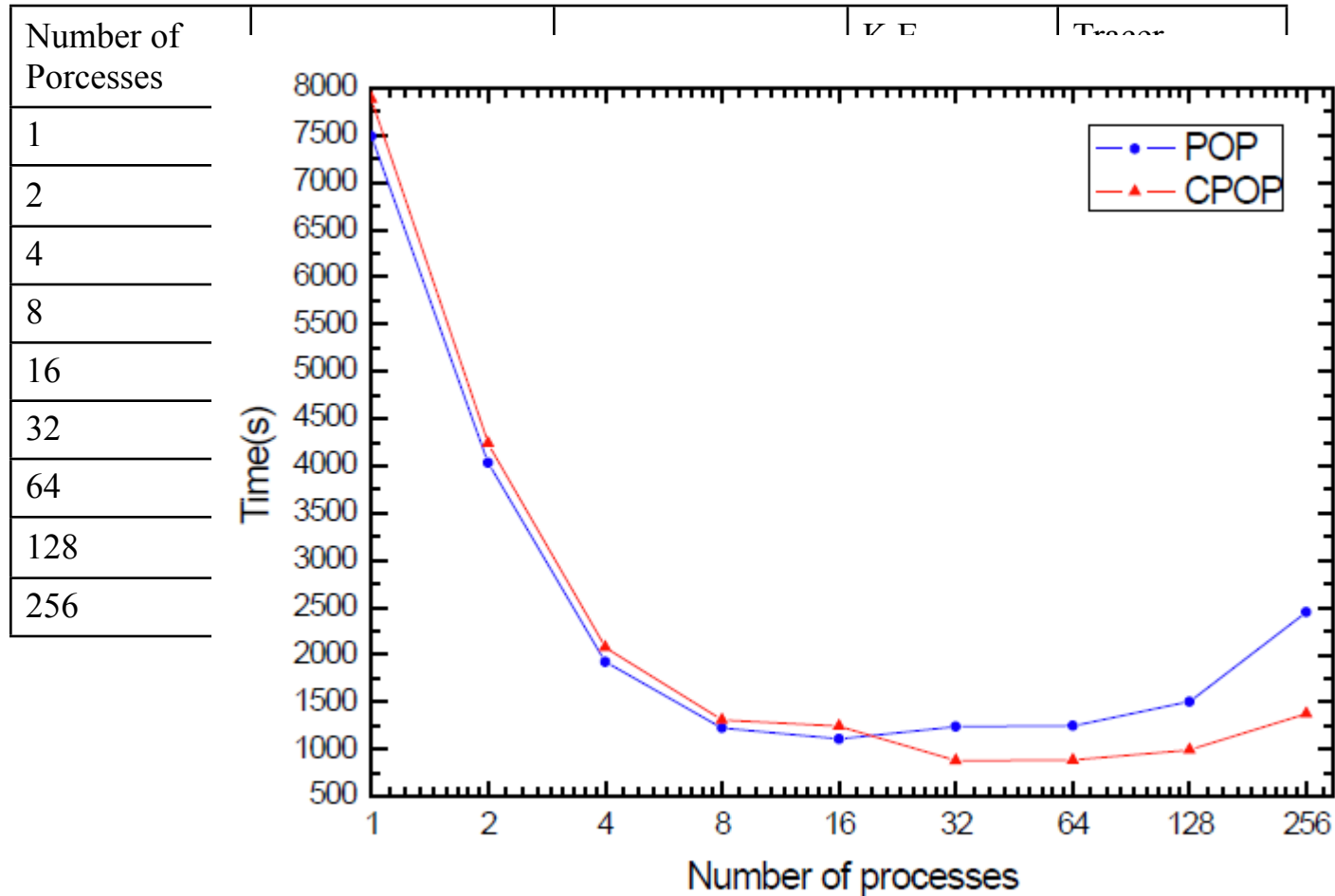
- Dynamic process organization based on grid partition





Validation and performance evaluation

➤ Functional validation and performance evaluation





Code generation

The screenshot displays a software development environment with the following components:

- Package Explorer (Left):** A tree view showing a project named 'POP' with sub-packages like 'POPstep_GridComp', 'POPdynamic_GridComp', 'POPbaroclinic_GridComp', etc.
- Diagram (Center):** A hierarchical diagram of code components. The root is 'POPstep_GridComp', which contains 'compInit', 'compRun', and 'compFinal'. It branches into 'POPdynamic_GridComp' and 'POPphysics_GridComp'. 'POPdynamic_GridComp' further branches into 'POPbaroclinic_GridComp', 'POPbarotropic_GridComp', 'POPhorizontal_GridComp', and 'POPvertical_GridComp'. 'POPphysics_GridComp' branches into 'POPsurface_GridComp' and 'POPsurface_CplComp'. 'POPsurface_GridComp' branches into 'surface_apComp', 'surface_pt_interiorComp', 'surface_s_interiorComp', 'surface_sfwfComp', 'surface_wsComp', and 'surface_compFinal'. 'POPbaroclinic_GridComp' branches into 'POPbaroclinic_CplComp' and 'POPbaroclinic_GridComp'. 'POPbaroclinic_CplComp' contains 'user_init1', 'user_init2', 'user_run1', 'user_run2', and 'user_final'. 'POPbaroclinic_GridComp' contains 'compInit', 'compRun', and 'compFinal'. 'POPbarotropic_GridComp' contains 'compInit1', 'compInit2', 'compRun', and 'compFinal'. 'surface_apComp' contains 'my_init', 'set_run1', 'restart_run2', and 'my_final'. 'surface_pt_interiorComp' contains 'my_init', 'set_run1', 'get_run2', 'restart_run3', and 'my_final'. 'surface_s_interiorComp' contains 'my_init', 'set_run1', 'set_run2', 'restart_run3', and 'my_final'. 'surface_sfwfComp' contains 'my_init', 'set_run1', 'restart_run2', and 'my_final'. 'surface_wsComp' contains 'my_init', 'my_run', and 'my_final'. 'surface_compFinal' contains 'my_init', 'set_run1', 'restart', and 'my_final'.
- Module Repository (Right):** A palette showing a list of available components and their icons, including 'POPstep_CplComp', 'POPice_GridComp', 'POPstep_Gri...', 'POPbaroclini...', 'POPdynamic_...', 'POPphysics_...', 'POPbarotrop...', 'POPhorizont...', and 'POPvertical_...'. Below the list are 'Tools' like 'AppDriver', 'GridComp', 'CplComp', 'Subroutine', 'RootComp', 'ChildGrid', and 'GridCplComp'.
- Module Property (Bottom):** A properties window for the 'AppDriver' component. It shows a table of properties:

Core	Property	Value
Appearance	Name	POP
	Root Comp	Grid Component POPstep_GridComp



Component specification

- Specification for component encapsulation
 - ◆ Templates to guide component development
 - ◆ Each component is associated with a xml-based description file before importing to the component library

◆ GridComponent		
Core	Property	Value
Appearance	Cdl	POPsurface_GridComp
	Description	POPsurface_GridComp
	Export State	
	Impl	POPsurface_GridComp
	Import State	
	Module Name	surfaceCompMod
	Name	surfaceComp
	States	shfexp,apimp,s_inimp,pt_inexp,wsimp,sfwfexp,sfwfimp,s_inexp,apexp,shfimp,wsexp,pt_inimp





Code generation (cont'd)

- Drag-Drop based component usage
 - ◆ Automatically locate and intercept the information about the component based on CDL, and then fill in the panel of attributes about this component
 - ◆ Component substitution can be done easily through selection

◆ GridComponent

Core	Property	Value
Appearance	Cdl	POPsurface_GridComp
	Description	POPsurface_apGridComp
	Export State	POPsurface_GridComp
	Impl	POPsurface_pt_interiorGridComp
	Import State	POPsurface_s_interiorGridComp
	Module Name	surfaceCompMod
	Name	surfaceComp
	States	shfexp,apimp,s_inimp,pt_inexp,wsimp,sfwfexp,sfwfimp,s_inexp,apexp,shfimp,wsexp,pt_inimp





Code generation (cont'd)

ESM Development - POP/src/surfaceComp/surfaceComp.F90 - Eclipse

File Edit Refactor Navigate Search Project External Run Saros Window Help

Fortran Projects

- surface_wsComp
- surfaceComp.F90
- GNUmakefile
- POPsurface_GridComp.xml
- surfaceComp.do
- POPTopStress_GridComp
- physicsComp.F90
- GNUmakefile
- physicsComp.do
- POPstep_CplComp
- stepComp.F90
- GNUmakefile
- stepComp.do
- popComp.F90
- GNUmakefile
- popComp.do
- src
 - surfaceComp
 - surfaceComp.F90
- .cproject
- .project

Remote Systems

- Local
 - Local Files
 - Local Shells

Outline

<Free Form - C Preprocessed>

- surfaceCompMod
 - surfaceComp_SetVM
 - surfaceComp_SetServices
 - my_init
 - my_run1
 - my_run2
 - my_final
- Syntax Errors

History

Saros Remote Screen

Not connected

No Session Running Buddies

To share projects you can either:

- 1) Right-click on a project
- 2) Right-click on a buddy
- 3) Use the Saros menu in the Eclipse menu bar

Console Problems Terminals SVN 资源库 Properties

Property	Value
name	my_init

```
end subroutine surfaceComp_SetServices

subroutine my_init(gcomp, importState, exportState, clock, rc)
type(ESMF_GridComp) :: gcomp
type(ESMF_State) :: importState
type(ESMF_State) :: exportState
type(ESMF_Clock) :: clock
integer, intent(out) :: rc

type(ESMF_Grid) :: parentgrid
type(ESMF_DistGrid) :: pareDistGrid
type(ESMF_Array)::SMF_array,SMFT_array,STF_array,SMF_array,SMFT_array,STF_array,TFW_array,ATM_P
call ESMF_StateAdd(wsimp,SMF_array,rc=rc)
call ESMF_StateAdd(wsimp,SMFT_array,rc=rc)
call ESMF_GridCompInitialize(ws_Comp, importState=wsimp, exportState =wsexp,phase=1,clock=pare
call ESMF_AttributeGet(wsexp, name="lsmft_avail", value =lsmft_avail, rc=rc)
call ESMF_StateAdd(shfexp,STF_array,rc=rc)
call ESMF_GridCompInitialize(shf_Comp, importState=shfimp, exportState =shfexp,phase=1,clock=pare
call ESMF_CplCompInitialize(surfaceCplComp, importState=pt_inexp, exportState =s_inimp,phase=1,
call ESMF_GridCompInitialize(sfwf_Comp, importState=sfwfimp, exportState =sfwfexp,phase=1,clock
call ESMF_GridCompInitialize(pt_in_Comp, importState=pt_inimp, exportState =pt_inexp,phase=1,c
call ESMF_GridCompInitialize(s_in_Comp, importState=s_inimp, exportState =s_inexp,phase=1,clock
call ESMF_GridCompInitialize(ap_Comp, importState=apimp, exportState =apexp,phase=1,clock=pare
call ESMF_StateAdd(exportState,wsexp,rc=rc)
call ESMF_StateAdd(exportState,shfexp,rc=rc)
call ESMF_StateAdd(exportState,sfwfexp,rc=rc)
call ESMF_StateAdd(exportState,pt_inexp,rc=rc)
call ESMF_StateAdd(exportState,s_inexp,rc=rc)
call ESMF_StateAdd(exportState,apexp,rc=rc)
call ESMF_StateAdd(exportState,SMF_array,rc=rc)
call ESMF_StateAdd(exportState,SMFT_array,rc=rc)
call ESMF_StateAdd(exportState,STF_array,rc=rc)
call ESMF_StateAdd(exportState,TFW_array,rc=rc)
call ESMF_StateAdd(exportState,ATM_PRESS_array,rc=rc)
call ESMF_StateAdd(exportState,FWL_array,rc=rc)
```

Component
Skeleton
Code





Module auto-verification

➤ Primary auto-verification for module validity

The screenshot shows the Eclipse IDE interface for an ES development project. The main window displays a module diagram for 'pop.model_diagram'. The diagram includes several modules and their relationships. A red dashed box highlights a specific module, 'POPsurface_GridComp', which is connected to other modules. An error dialog box is open, displaying the message: 'surfaceComp should contain a cpl named POPsurface_CplComp'. The text 'Coupler not Match' is overlaid on the diagram, indicating a mismatch in the coupler configuration. The Properties view at the bottom shows the details for the selected 'GridComponent', including its core, appearance, and various properties like Cdl, Description, Export State, Impl, Import State, Module Name, Name, and States.

Core	Property	Value
Core	Cdl	surfaceComp_cpl
	Description	POPsurface_GridComp
Appearance	Export State	
	Impl	surfaceComp_cpl
Core	Import State	
	Module Name	surfaceCompMod
	Name	POPsurface_GridComp
	States	



Model components substitution

- Model component encapsulation allows the components developed by different groups or communities to be inter-changeable, reducing duplicated programming efforts
 - ◆ Component substitution is demonstrated between POP and MOM
 - ◆ MOM
 - a three-dimensional ocean circulation model designed primarily by the Geophysical Fluid Dynamics Laboratory (GFDL) of the National Oceanic and Atmospheric Administration
 - ◆ POP
 - developed and supported by Los Alamos National Laboratory (LANL)



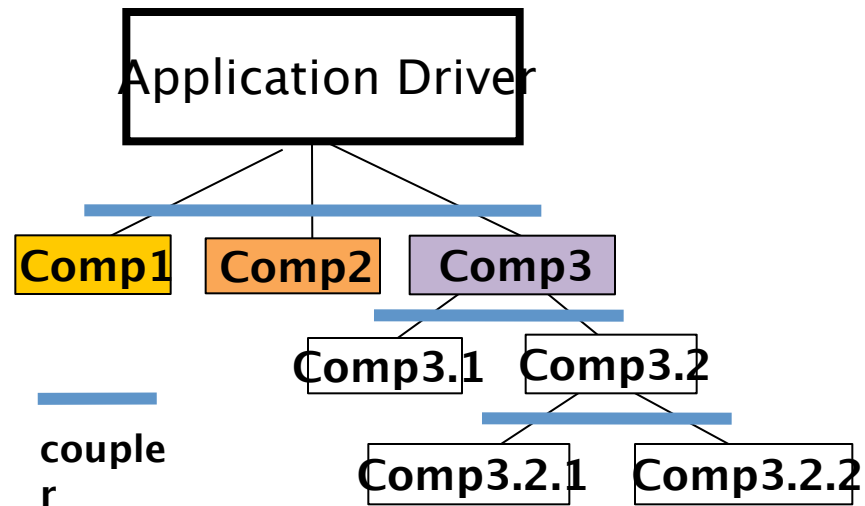


POP vs MOM

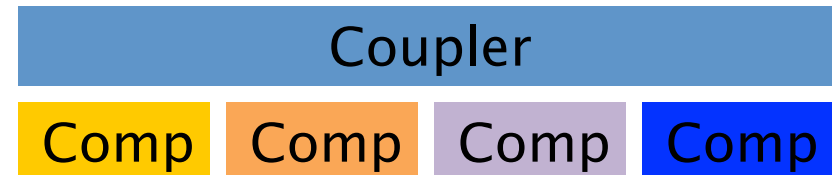
➤ Framework Comparison

- ◆ POP is based on ESMF, while MOM is based on FMS

ESMF



FMS



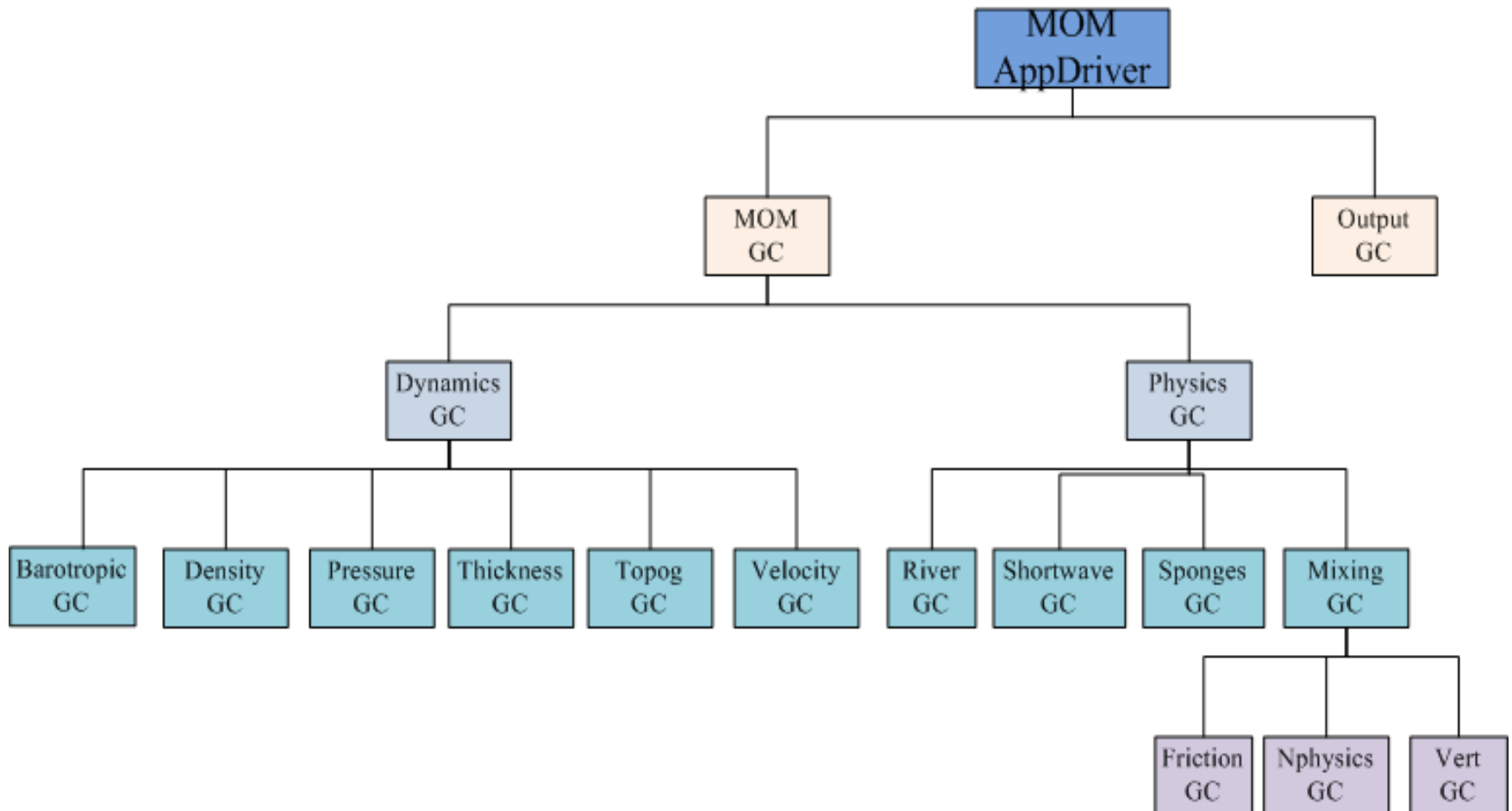
- Components are executed sequentially
- All component resides in flat
- Single coupler
- Data is transferred through shared public variables

- Child components are executed simultaneously
- Invocation among components is nested
- Multiple couplers
- Data is transferred through data structure called state





MOM code structure analysis





MOM code structure analysis

- Analyzing the code structure of MOM
 - ◆ MOM component interface
 - init() #initialize and configure
 - run()
 - ◆ update() #update data, time step, status and so on
 - ◆ restart() #reload configuration and restart
 - final(): #release the resouces

The above interface would make sure that the components derived from MOM, which is based on FMS specification, are compatible with components derived from POP which is based on ESMF, so that the components are substitutable





Outline



- Motivation
- Purpose and Significance
- Research Contents
- Technology Roadmap
- IDE design & development functions
- ESM Module Encapsulation
- Runtime Support**
- Deployment





Runtime support

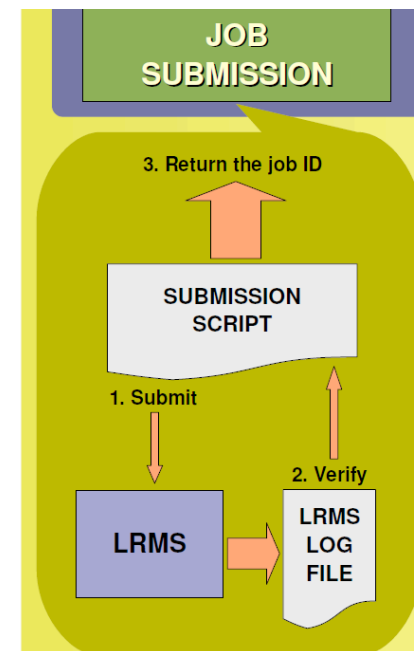
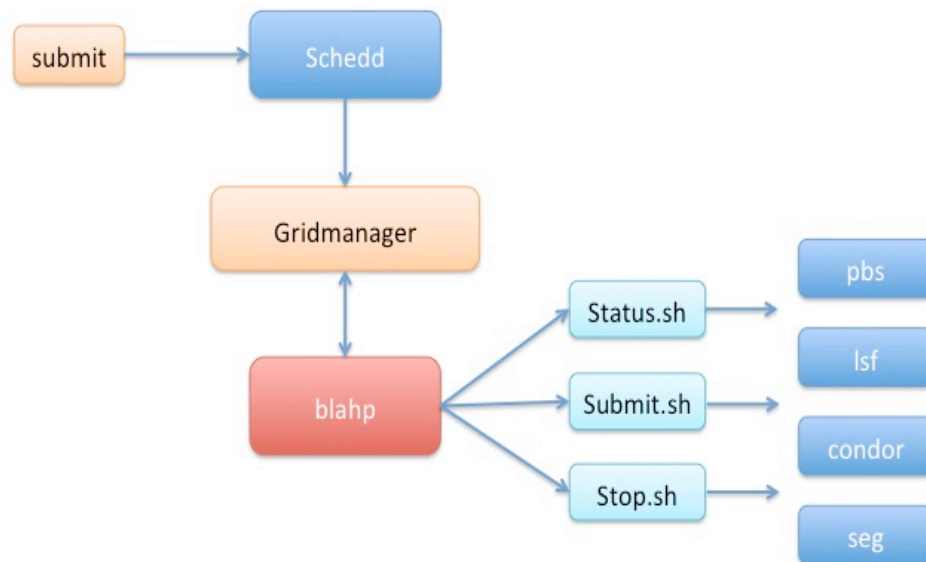
- Work have been done :
 - ◆ Seamless integration with Slurm batch system with an easy-to-use management portal
 - ◆ Support ESM program execution monitoring and trace analysis





Integration with the batch system

- Integration with Condor to provide **persistent execution environment**
- The jobs submitted to the IDE are delegated to underlying slurm batch system
 - Extend condor to talk to slurm
 - Submit, query and management





Program execution monitoring

ESM Development - pop/source/diagnostics.F90 - Eclipse Platform

File Edit Refactor Navigate Search Project Run Window Help

Resource Monitor

Project Explorer: BNU ESM.new

NodeName	Arch	OS	Sockets	State	Reason
node1	x86_64	Linux	8	2	
node2	x86_64	Linux	16	2	
node3	x86_64	Linux	16	2	
node4	x86_64	Linux	16	2	
node5	x86_64	Linux	16	2	

Job Submit

submit a job:

job's name: pop

min nodes need: 8

number of tasks: 4

script: srun /root/mount/yes/pop/Application

std_out: pop.out

Job Query

JobID	Name	JobState	Submit_time	Start_time	End_time
767	ping	JOB CANCELED	2011-06-30 15:21:29	2011-06-30 15:21:29	2011-06-30 15:21:29
768	pop	JOB COMPLETED	2011-06-30 16:47:13	2011-06-30 16:47:13	2011-06-30 16:47:13

```
use avg, only: define_avg_fie
use forcing, only: STF, SMF, lsr
use forcing_shf, only: SHF_QSW,
! use forcing_coupled, only:
use forcing_pt_interior, only: s
use forcing_s_interior, only: se
use exit_mod, only: sigAbort, e
```

修订	日期	作者	注释
*63	11-5-12 上午9...	ys	New imj

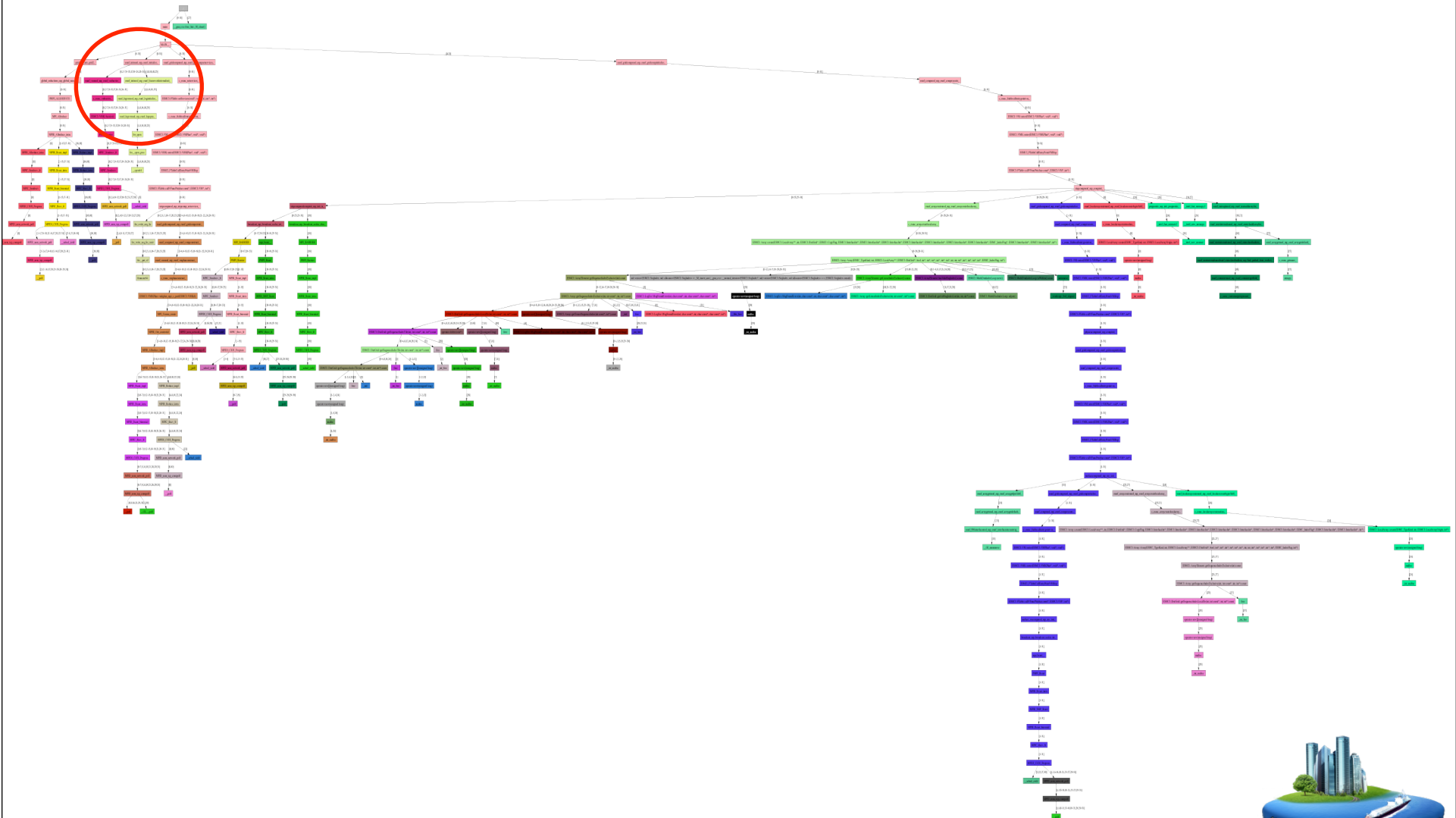
File: C:\setup.log

Command mode: _____

10:55
2011/5/13

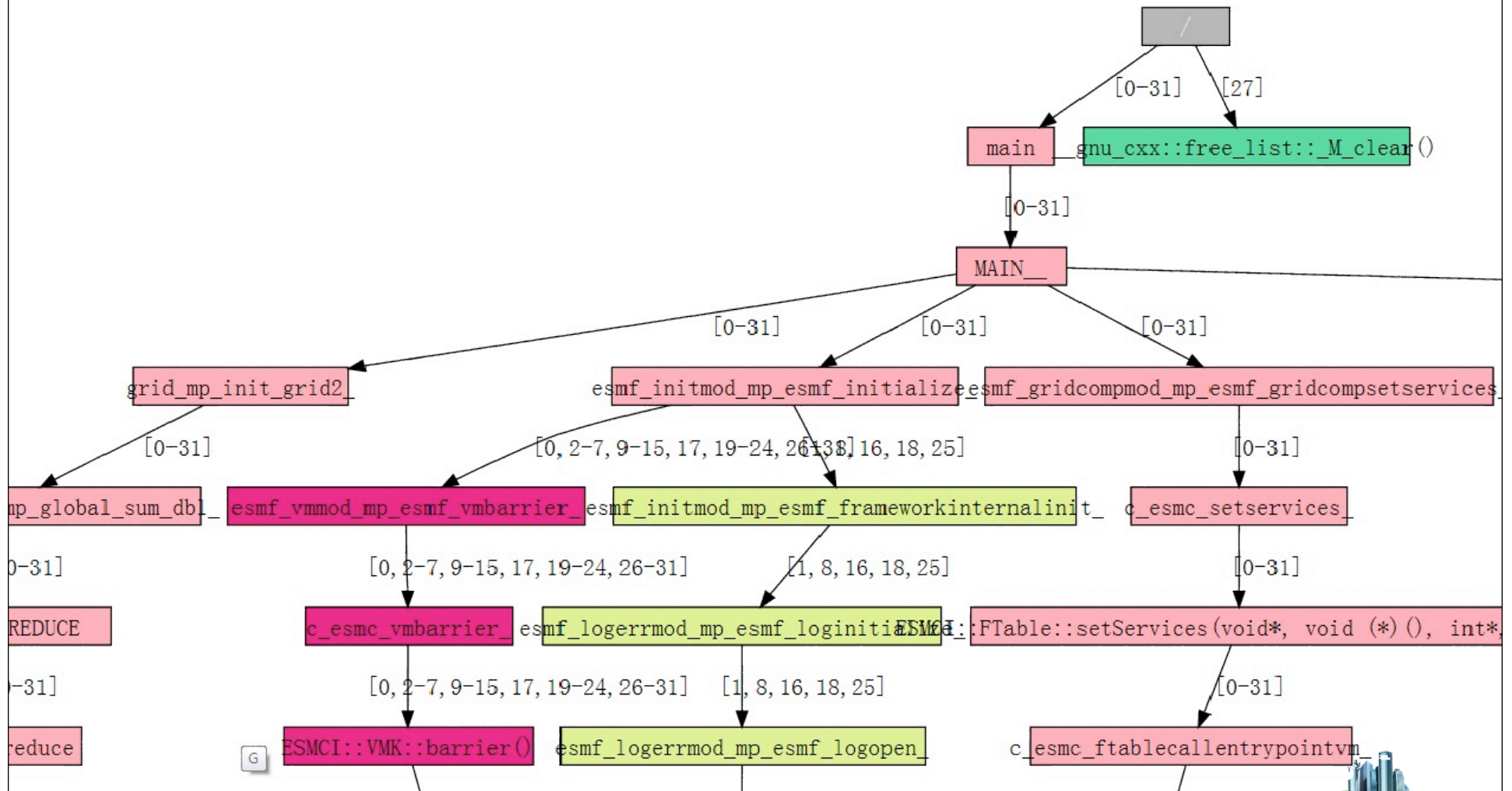


Trace analysis based on STAT





Trace analysis based on STAT (cont'd)





Outline





Deployment

- Alpha version released (March 2011)
- Five versions (v0.2, v0.4, v0.6, v1.0, v2.0) released so far
- Deployed in **Beijing Normal University** (since March 2011)
 - ◆ Pilot usage of four versions (v0.4, v0.6, v1.0, v2.0)
 - ◆ Used in BNU_ESM development
 - ◆ Latest version is v2.0
- Deployed in **Institute of Atmospheric Physics, Chinese Academy of Sciences** (since March 2011)
 - ◆ Latest version is v2.0
- Deployed in **National Climate Center** (since June 2011)
 - ◆ Latest version is v2.0
- IDE and model encapsulation approach adopted by **the coupler development project** (since July 2011)



Thank You!

