



# 地球系统数值模拟装置项目 (地球系统模式数值模拟系统) 大气环流模式分系统培训

培训人：曹杭

2022年6月1日

01

分系统介绍

---

02

程序代码结构

---

03

模式优化

---

04

新建case、编译、运行

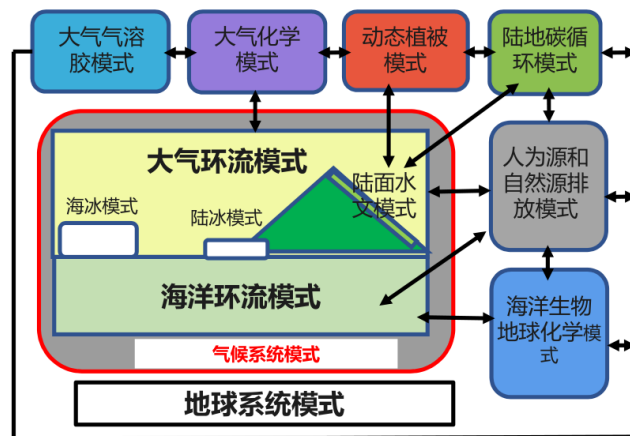
---



# 01 分系统介绍

## IAP-AGCM物理模型与特点:

- 模拟地球系统模式中的大气的大尺度运动过程
- 模拟大气温度、风速、压力、降水、云量、水汽等基本的大气环流特征
- 模式动力框架基于C网格有限差分离散



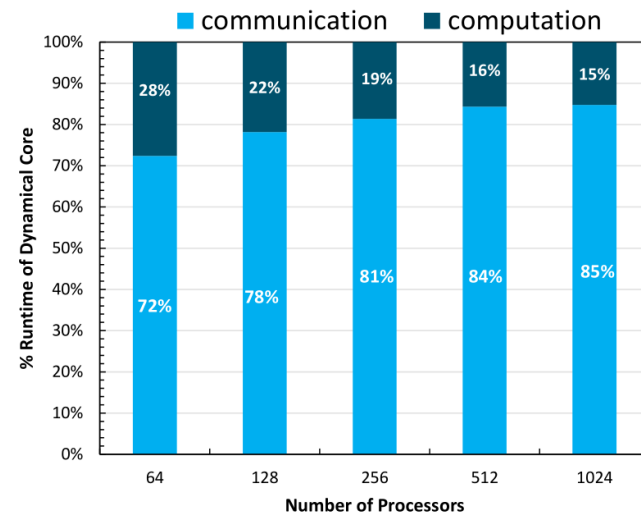
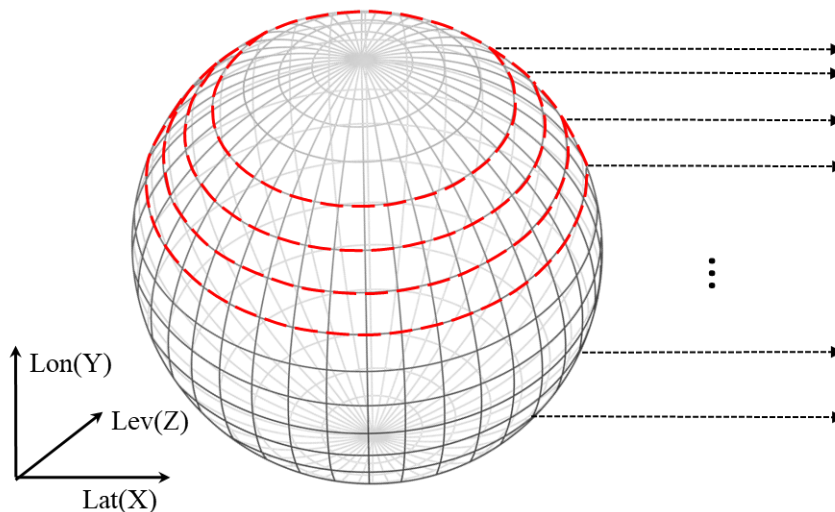
地球系统模式组成

$$\left\{ \begin{aligned} \frac{\partial U}{\partial t} &= -\sum_{m=1}^3 \alpha^* L_m(U) - \beta^* P_\lambda - \gamma^* f^* V \\ \frac{\partial V}{\partial t} &= -\sum_{m=1}^3 \alpha^* L_m(V) - \beta^* P_\theta + \gamma^* f^* U \\ \frac{\partial \Phi}{\partial t} &= -\sum_{m=1}^3 \alpha^* L_m(\Phi) + (1 - \delta_p) \\ &\quad \cdot [b(1 + \delta_c) + \delta \cdot \kappa \Phi / P] \cdot \beta^* \tilde{\Omega} \\ \frac{\partial}{\partial t} (p'_{sa}/p_0) &= -\beta^* \tilde{P}(W) + \kappa^* D_{sa}/P_0 \end{aligned} \right.$$

IAP-AGCM动力框架差分方程组

## 并行划分方式:

- 按照**经纬网格方向**划分
- 多维度的**MPI并行**实现
- 并行优化点主要为**通信优化**
- 物理过程适合异构加速



01

分系统介绍

---

02

程序代码结构

---

03

模式优化

---

04

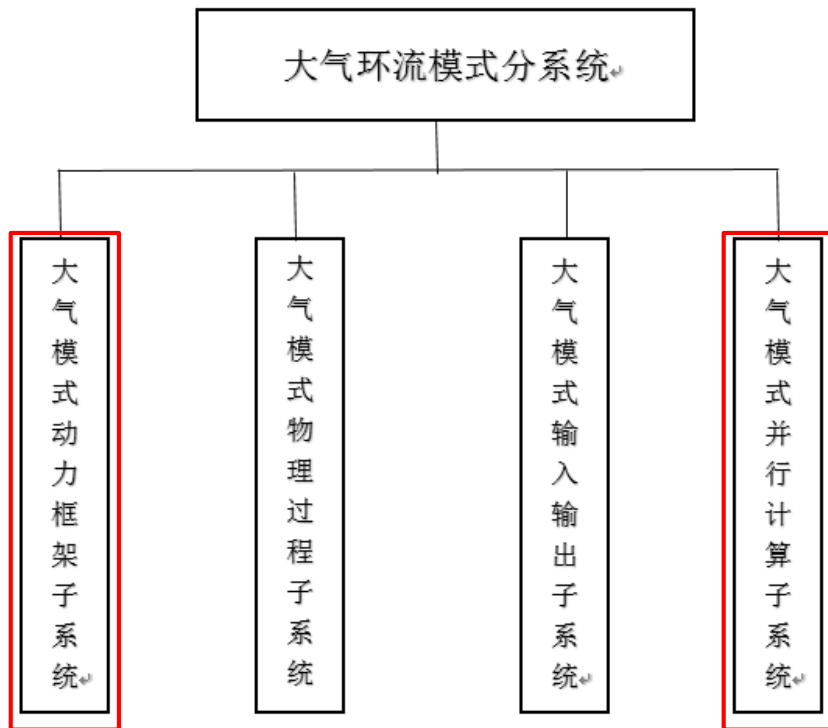
新建case、编译、运行

---

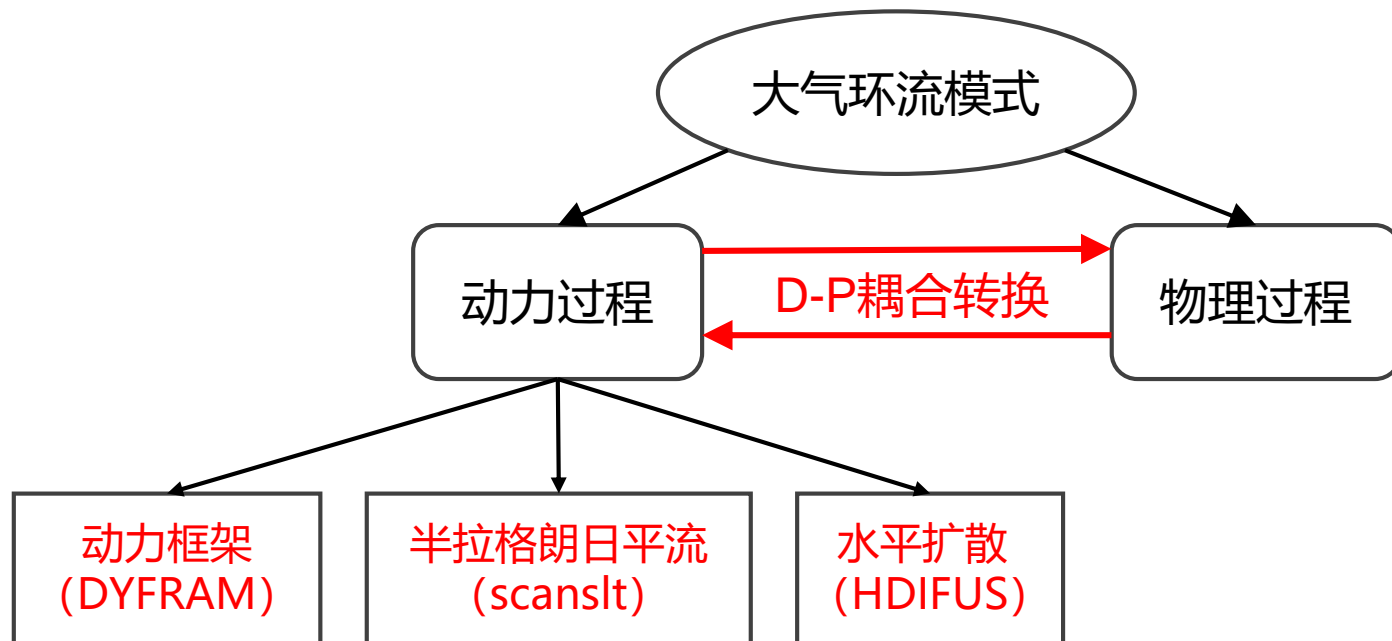


# 02 程序代码结构

## 大气环流模式分系统组成:

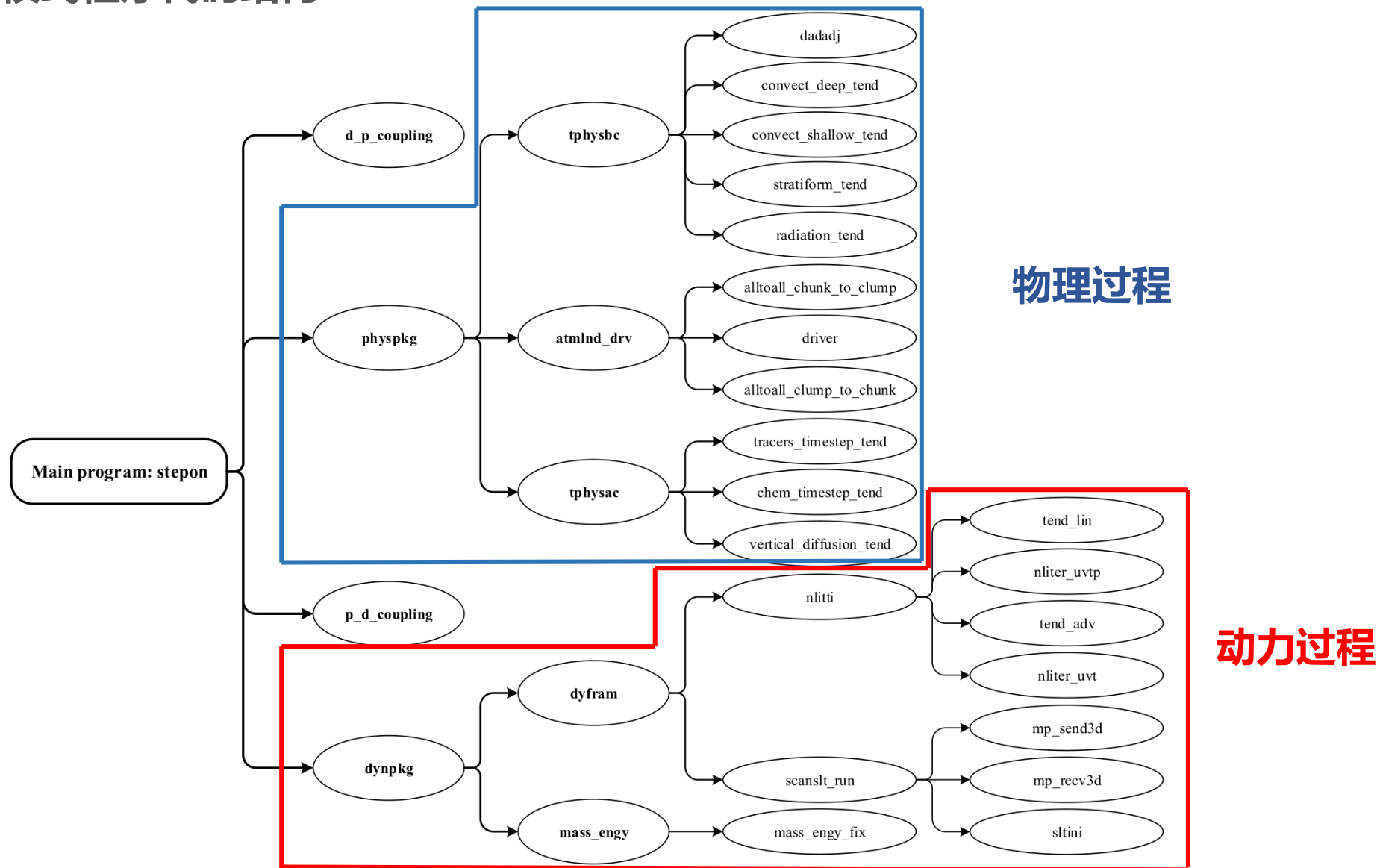


## 大气环流模式程序架构:



# 02 程序代码结构

## 大气环流模式程序代码结构:



01

分系统介绍

02

程序代码结构

03

模式优化

04

新建case、编译、运行



# 03 模式优化

## 大气模式并行优化的挑战:

- 应用的Stencil计算模式的通信量大
- 高分辨率下经纬网格划分方式导致的负载不均衡, 可扩展性和并行性能受限
- 高分辨模拟执行速度较慢

## 主要优化工作:

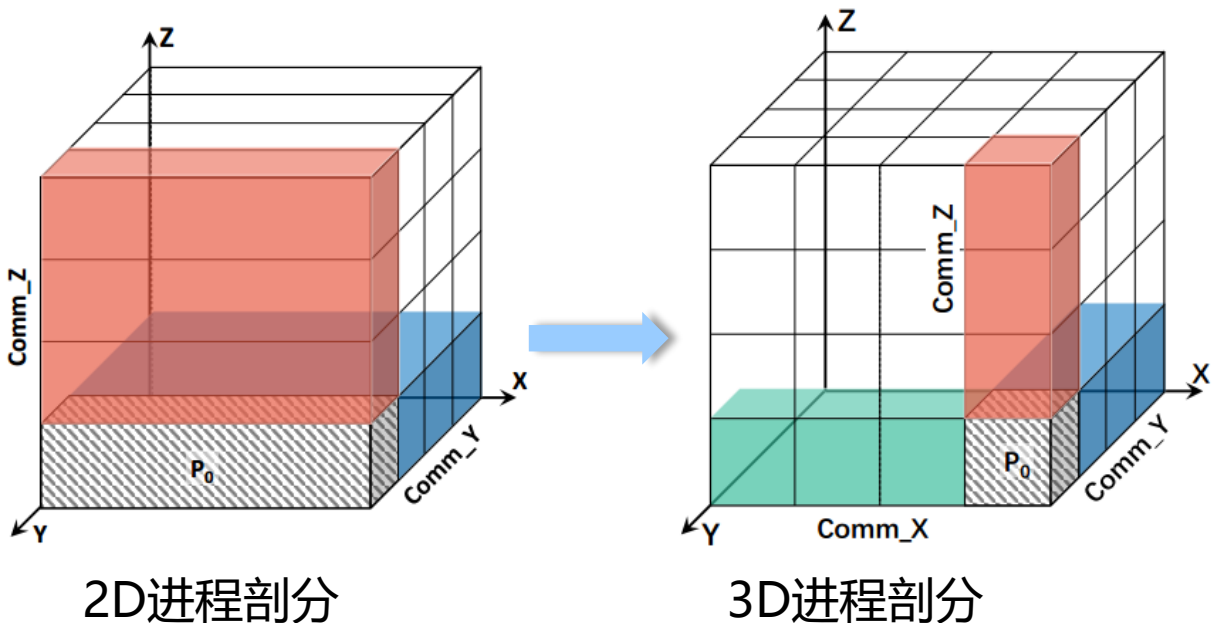
1. 三维并行剖分及优化
2. Z-prior性能优化
3. 高分辨率跳点差分优化
4. 耦合转换模块开发与改进
5. 通信与计算重构优化



# 03 模式优化

## 三维并行剖分及优化

- 传统的大气模式采用二维剖分的并行模式，随着分辨率提升，网格数增多，计算量增加，并行性能与可扩展性受限;
- 采用基于三维剖分的并行优化策略，提高并行效率和扩展性;
- 针对大气环流模式的纬向、经向和垂向计算网格，增加了纬向(x)剖分以提高并行度。



单进程计算网格点数:

$$\frac{M * N * H}{P_y * p_z}$$

↓

$$\frac{M * N * H}{P_x * P_y * p_z}$$

## 三维并行剖分模式集成

- 已将AGCM3D加入到地球系统模式CAS-ESM\_dev版本中，3D/2D动力框架可选；
- 已添加的可选compset包括有：FIDEAL、FADIAB、F\_2000等；
- 3D可选分辨率：f3d14、f3d05、f3d02
- 已上传git服务器 ict分支。

```
<compset NAME="F_IDEAL_PHYS" SHORTNAME="FIDEAL"
DESC="INVALID: AMIP runs not supported at this resolution"
COMP_ATM="cam" COMP_LND="slnd" COMP_ICE="sice" COMP_OCN="socn" COMP_GLC="sglc"
CCSM_CCOST="1"
CCSM_CO2_PPMV="284.7"
CAM_NML_USE_CASE="1850_cam5"
CAM_CONFIG_OPTS="-phys ideal" />

<compset NAME="F_IDEAL_PHYS" SHORTNAME="FIDEAL" GRID_MATCH="fd14_fd14"
DESC="H-S run with ideal physics" />

<compset NAME="F_IDEAL_PHYS" SHORTNAME="FIDEAL" GRID_MATCH="f3d14_f3d14"
DESC="H-S run with ideal physics (IAP_3D)" />

<compset NAME="F_IDEAL_PHYS" SHORTNAME="FIDEAL" GRID_MATCH="f3d05_f3d05"
DESC="H-S run with ideal physics (IAP_3D)" />

<compset NAME="F_ADIAB_PHYS" SHORTNAME="FADIAB" GRID_MATCH="f3d02_f3d02"
DESC="Adiabatic run without physics (IAP_3D)" />

<compset NAME="F_ADIAB_PHYS" SHORTNAME="FADIAB" GRID_MATCH="f3d05_f3d05"
DESC="Adiabatic run without physics (IAP_3D)" />

<compset NAME="F_ADIAB_PHYS" SHORTNAME="FADIAB" GRID_MATCH="f3d14_f3d14"
DESC="Adiabatic run without physics (IAP_3D)" />
```

```
<horiz_grid GRID="0.5x0.5_0.5x0.5" SHORTNAME="f3d05_f3d05"
ATM_GRID="0.5x0.5" LND_GRID="0.5x0.5" OCN_GRID="0.5x0.5" ICE_GRID="0.5x0.5"
ATM_NCPL="48" OCN_NCPL="48"
CAM_DYCORE="lap3d"
CCSM_GCOST="4"
MAP_A20F_FILE="idmap"
MAP_A20S_FILE="idmap"
MAP_O2AF_FILE="idmap"
MAP_O2AS_FILE="idmap"
MAP_A2LF_FILE="idmap"
MAP_A2LS_FILE="idmap"
MAP_L2AF_FILE="idmap"
MAP_L2AS_FILE="idmap"
MAP_A2WF_FILE="idmap"
MAP_A2WS_FILE="idmap"
MAP_W2AF_FILE="idmap"
MAP_W2AS_FILE="idmap"
MAP_A2GEF_FILE="idmap"
MAP_A2GES_FILE="idmap"
MAP_GE2AF_FILE="idmap"
MAP_GE2AS_FILE="idmap"

MAP_R20_FILE_R05="idmap"
MAP_R20_FILE_RX1="idmap" />
```

# 03 模式优化

## 大气模式并行优化的挑战:

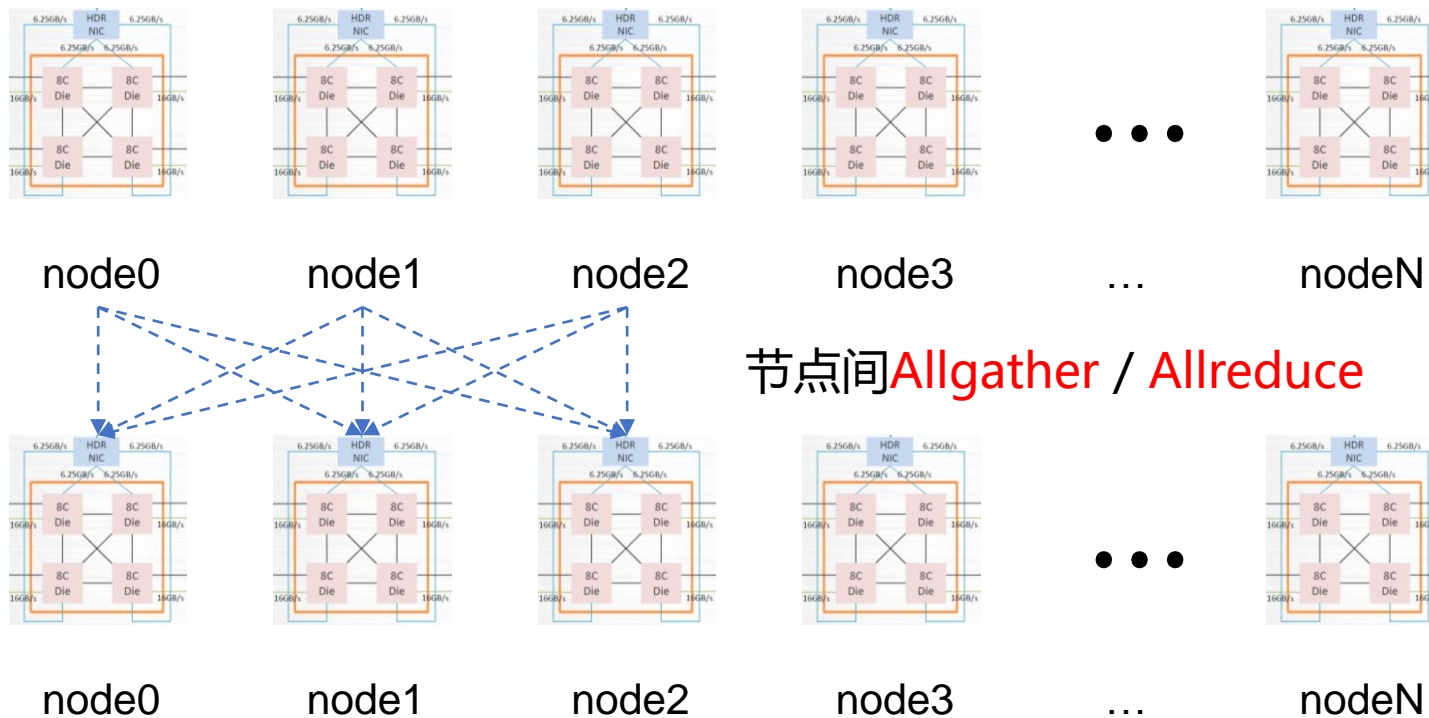
- 应用的Stencil计算模式的通信量大
- 高分辨率下经纬网格划分方式导致的负载不均衡, 可扩展性和并行性能受限
- 高分辨模拟执行速度较慢

## 主要优化工作:

1. 三维并行剖分及优化
2. Z-prior性能优化
3. 高分辨率跳点差分优化
4. 耦合转换模块开发与改进
5. 通信与计算重构优化

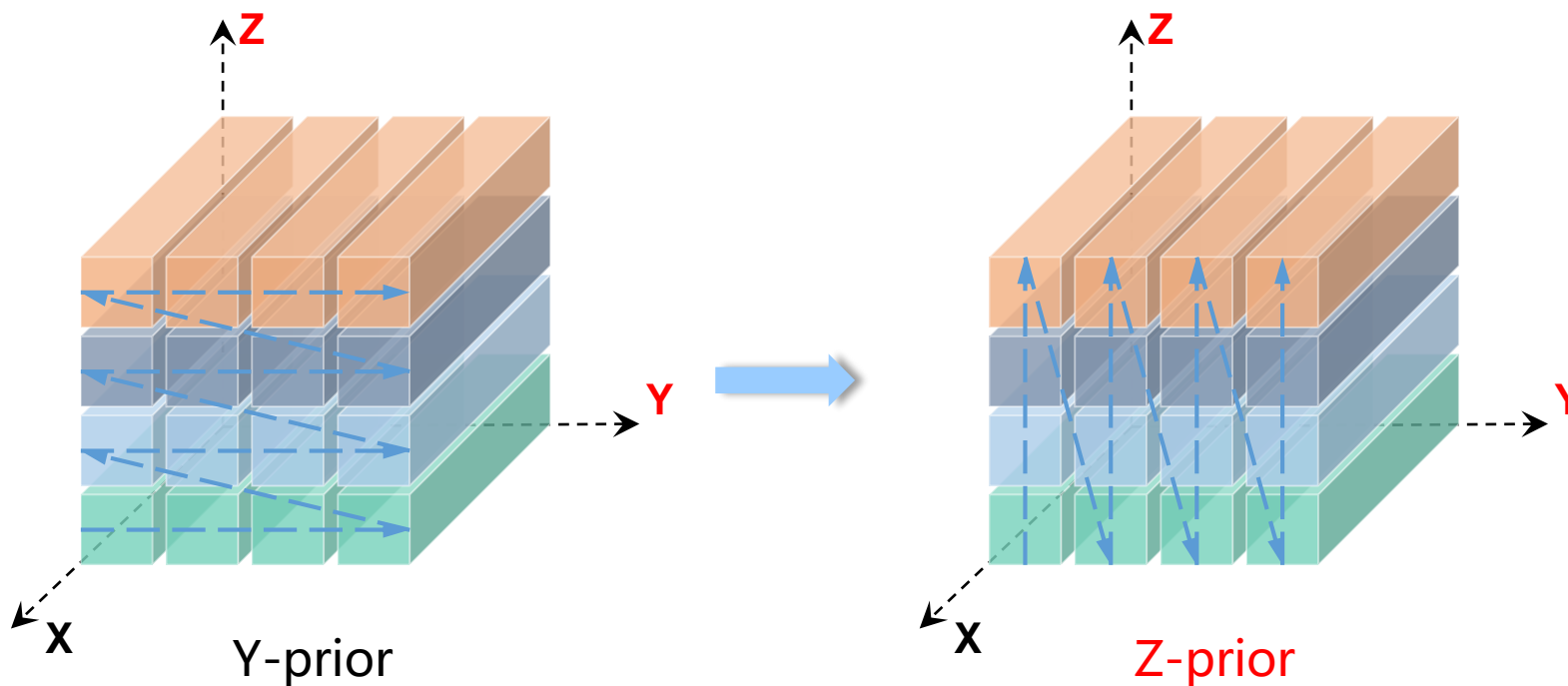
# 03 模式优化

- 大气模式原始进程剖分优先顺序为：Y-Z(2D)、X-Y-Z(3D)；
- 原始进程剖分顺序下，垂直方向(Z)通信为**节点间MPI通信**；
- 在适应过程和平流过程中，Z方向的**集合通信**和点到点通信开销随着Z方向进程增多而增大。



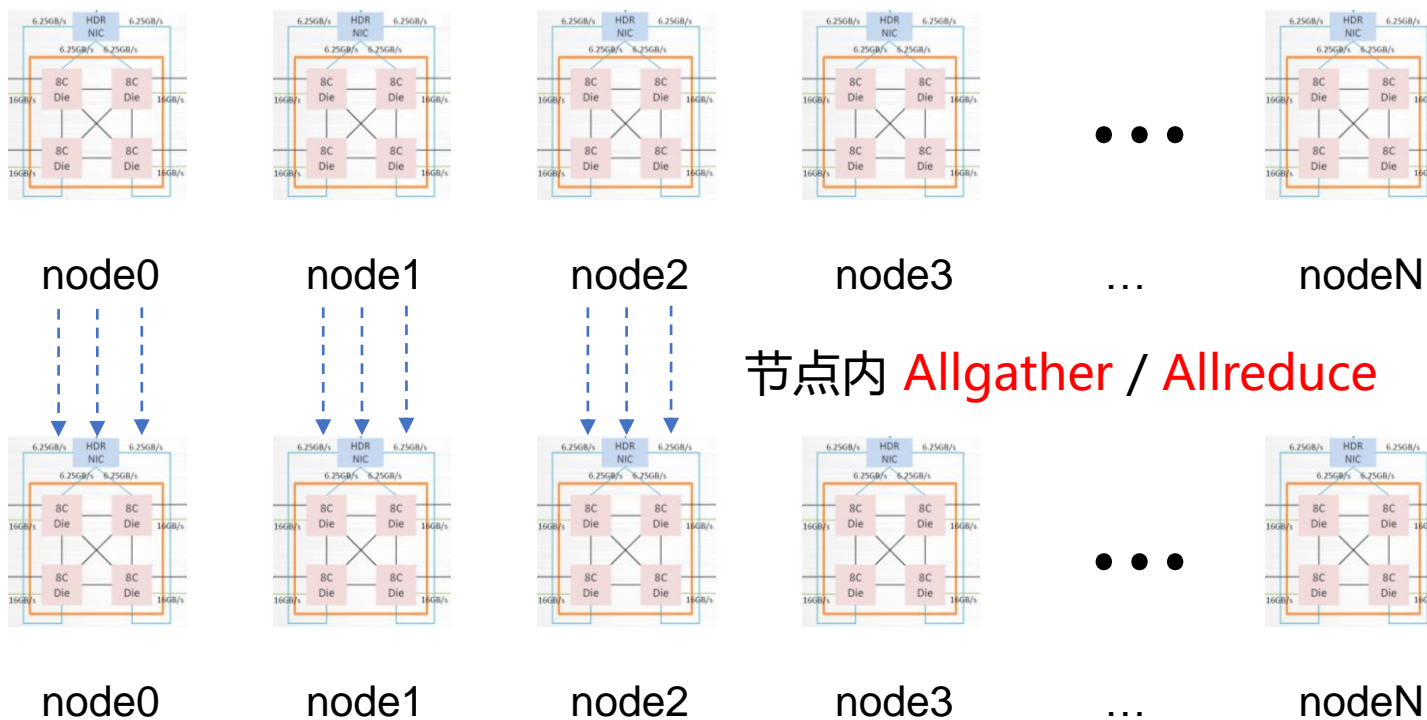
# 03 模式优化

- 将进程划分顺序修改为: Z-Y(2D)、Z-X-Y(3D);
- 垂直方向通信转换为节点内MPI通信;
- 集合通信开销大大降低。



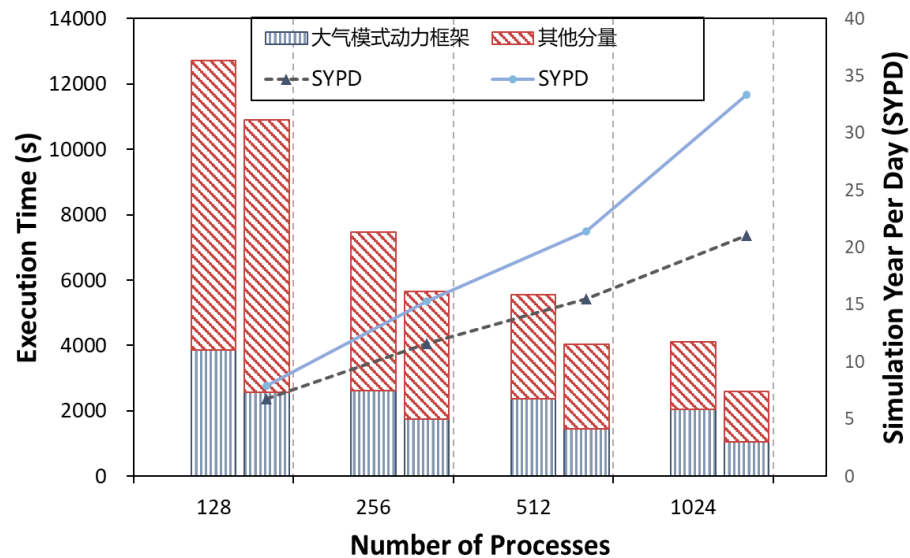
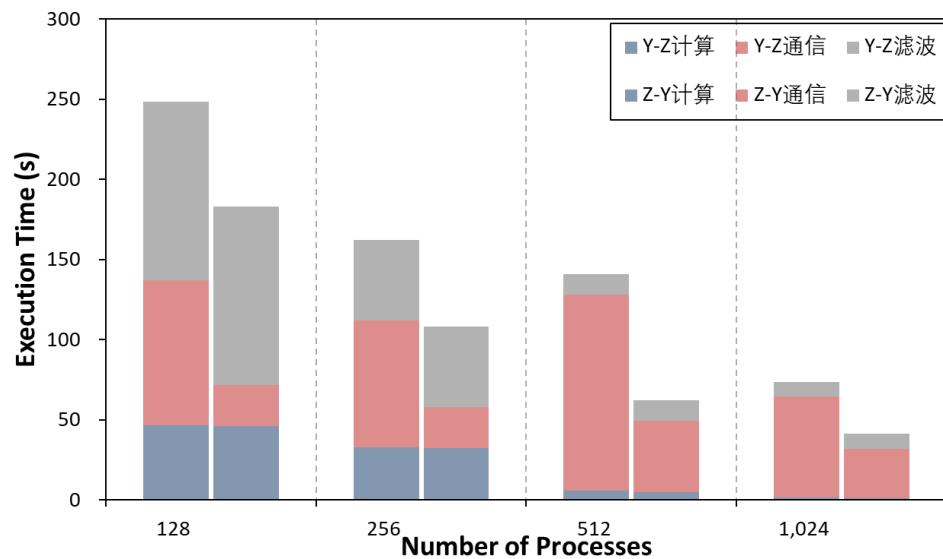
# 03 模式优化

- 将进程划分顺序修改为: Z-Y(2D)、Z-X-Y(3D);
- 垂直方向通信转换为节点内MPI通信;
- 集合通信开销大大降低。



# 03 模式优化

- 二维动力框架版本,  $1.4^{\circ} \times 1.4^{\circ}$ , 128-1024进程, 1month模拟, 性能提升如下:
  - 通信时间平均减少63.6%;
  - 动力框架执行时间平均减少39.8%。



# 03 模式优化

## 大气模式并行优化的挑战:

- 应用的Stencil计算模式的**通信量大**
- 高分辨率下经纬网格划分方式导致的**负载不均衡**,  
可扩展性和并行性能受限
- 高分辨模拟执行速度较慢

## 主要优化工作:

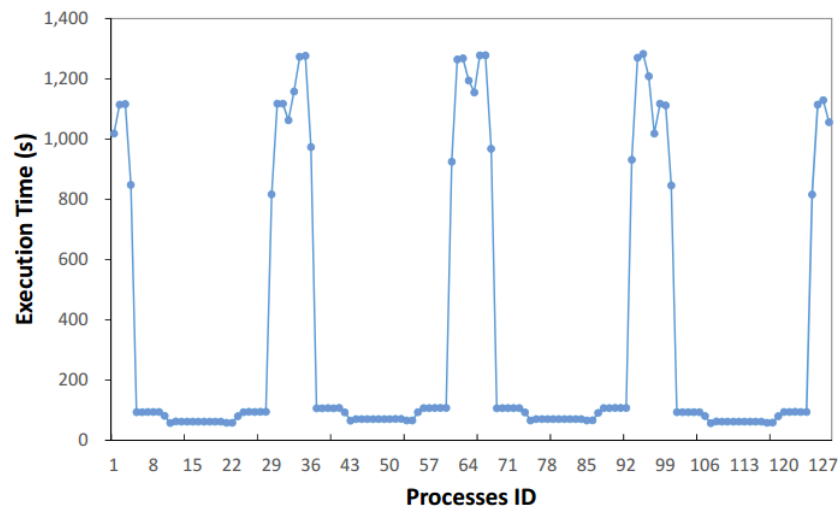
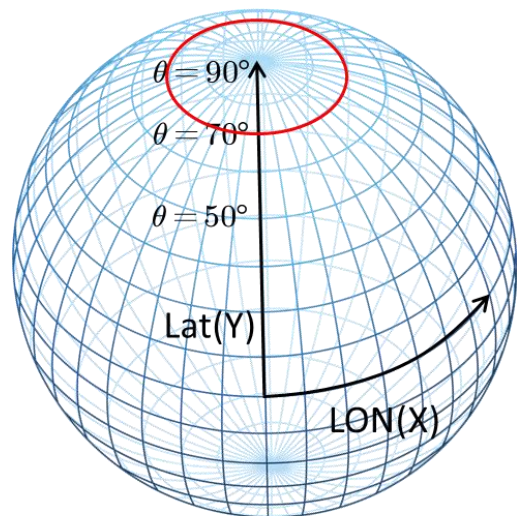
1. 三维并行剖分及优化
2. Z-prior性能优化
3. **高分辨率跳点差分优化**
4. 耦合转换模块开发与改进
5. 通信与计算重构优化



# 03 模式优化

## ● “极点”问题

- 对于IAP-AGCM, **经纬网格**离散化方案使得极区的物理格距急剧变小, 产生纬向短波。因而针对模式中变量的计算, 在极区需要采用**纬向滤波**模块以保证计算稳定性。
- 而随着大气模式**分辨率提升**, 网格数增多, 计算量增加, 极区的纬向多次滤波模块使大气模式的**并行性能与可扩展性**受限。
- 高纬滤波模块与中低纬度区域的计算不匹配会引起严重的**负载不均衡**。

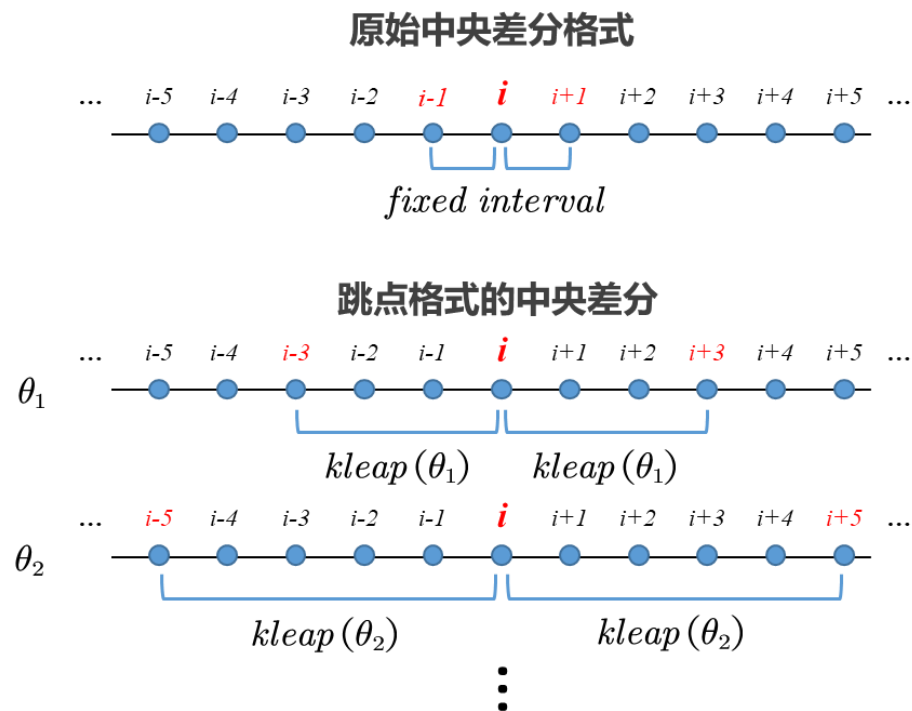


各进程执行时间 (128进程)

# 03 模式优化

## ● 纬向自适应跳点差分

- A. 在中低纬度位置，保持原滤波模块不变;
- B. 在高纬位置 ( $\geq 70^\circ$ )，使用跳点差分格式替代FFT滤波;
- C. 跳点格距计算以 $45^\circ$ 位置为参考阈值，随纬度增加自适应变化。



### 原始中央差分格式

$$\left( \frac{\partial F}{a \sin \theta \partial \lambda} \right)_{i,j} = \frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{a \sin \theta_j \Delta \lambda}$$

### 跳点格式的中央差分：(高纬)

$$\left( \frac{\partial F}{a \sin \theta \partial \lambda} \right)_{i,j} = \frac{F_{\text{leap}1,j} - F_{\text{leap}2,j}}{a \sin \theta_j \Delta \lambda * 2 * (\text{k leap} - 1)}$$

$$\text{k leap} = \frac{\arcsin(\cos 45^\circ \times \sin 0.5^\circ)}{(N \times \arcsin(\cos \text{lat} \times \sin 0.5^\circ))}$$

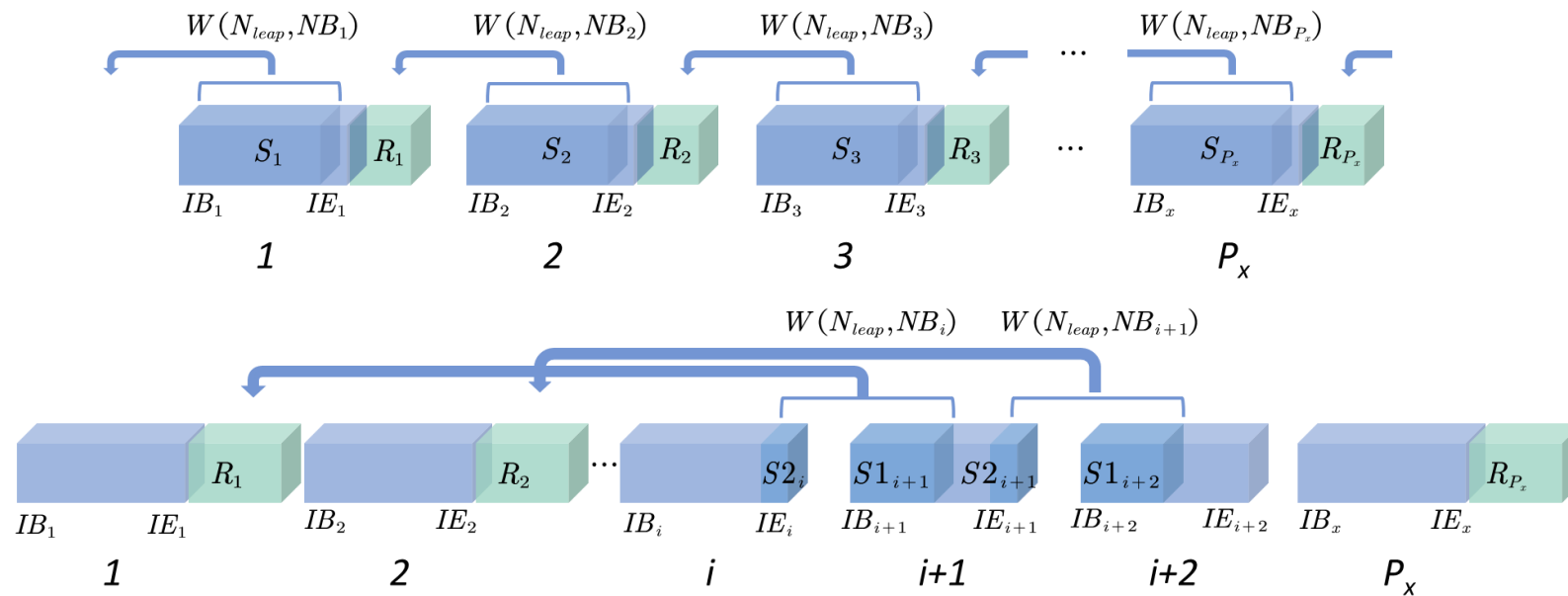
$$\text{leap}1 = \text{mod}(I + \text{k leap}, NX) + \frac{I + \text{k leap}}{NX \times (IB + 2)}$$

$$\text{leap}2 = \frac{\text{k leap} - 1 + NLON}{I + NLON} \times NLON + (I - \text{k leap} + 1)$$

# 03 模式优化

## ● 跳点差分三维并行化

- A. 按照高纬位置 ( $\geq 70^\circ$ ) 每个纬圈的跳点个数, 确定不同的纬向通信方案;
- B. 针对极区部分的较小的跳点数, 需要进行邻居进程的循环通信;
- C. 针对极区部分的较大跳点数, 需要进行跨多进程的循环通信;
- D. 跨多进程通信处理时, 需要注意跨进程的周期性边界。



# 03 模式优化

## 大气模式并行优化的挑战:

- 应用的Stencil计算模式的**通信量大**
- 高分辨率下经纬网格划分方式导致的**负载不均衡**,  
可扩展性和并行性能受限
- 高分辨模拟执行速度较慢

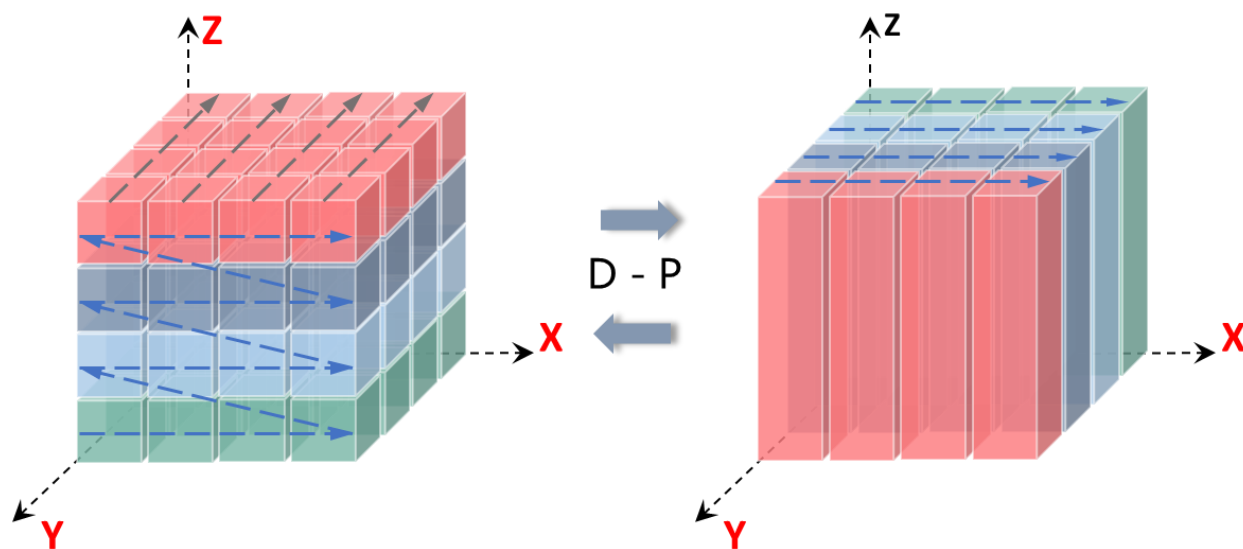
## 主要优化工作:

1. 三维并行剖分及优化
2. Z-prior性能优化
3. 高分辨率跳点差分优化
4. **耦合转换模块开发与改进**
5. 通信与计算重构优化

# 03 模式优化

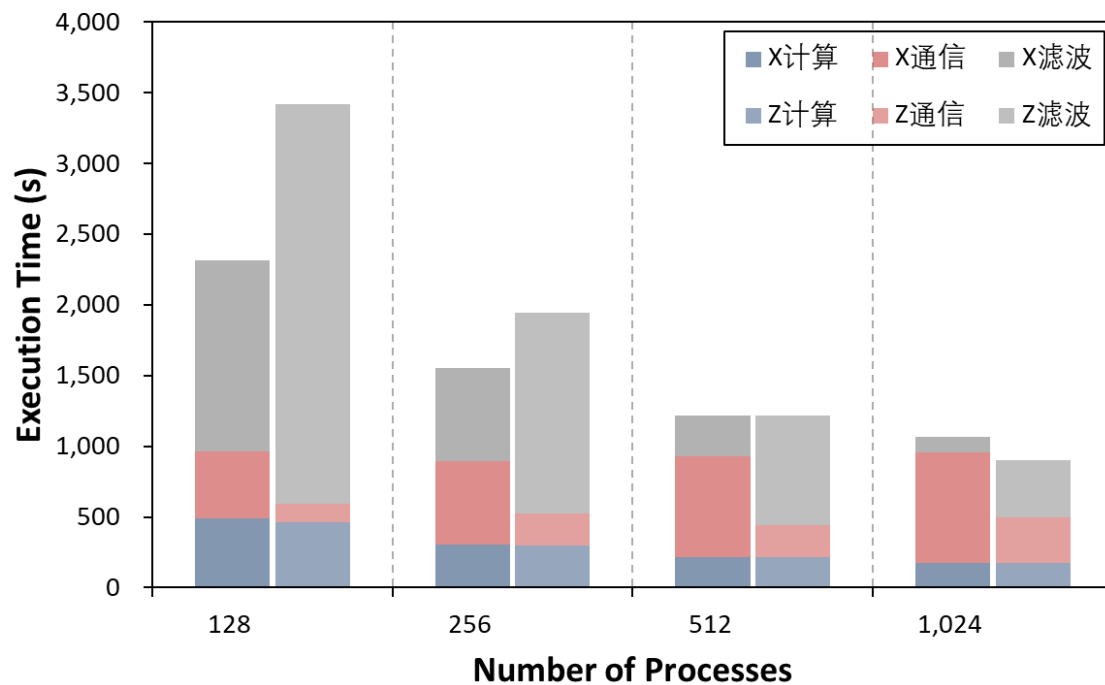
## ● “动力-物理” 转换耦合

- 为了最大程度提高模式的并行效率，大气模式的物理过程和动力框架采用了不同的数据结构，动力框架与物理过程数据结构之间需要相互转化耦合；
- 针对D-P转换耦合模块，完成了Z-prior进程划分物理过程和三维动力框架数据结构之间的转化的相适应的修改及验证；
- 以X-Y-Z和X-Y通信子域的通信完成D-P转换，同时结合与二维动力框架相似的剖分优化策略，减少D-P转换的开销。



# 03 模式优化

- 针对三维动力框架版本，进行了 $0.5^\circ \times 0.5^\circ$ ，128-1024进程，10模式天的测试，模拟结果如下：
  - 总通信时间平均减少65%；
  - D-P耦合转换时间平均减少约10%；
  - 由于三维动力框架的影响，在较小规模下，x方向通信开销会因为进程剖分优先顺序调节而增加；在更大规模下，Z-prior会有优势。



# 03 模式优化

## 大气模式并行优化的挑战:

- 应用的Stencil计算模式的**通信量大**
- 高分辨率下经纬网格划分方式导致的**负载不均衡**,  
可扩展性和并行性能受限
- 高分辨模拟执行速度较慢

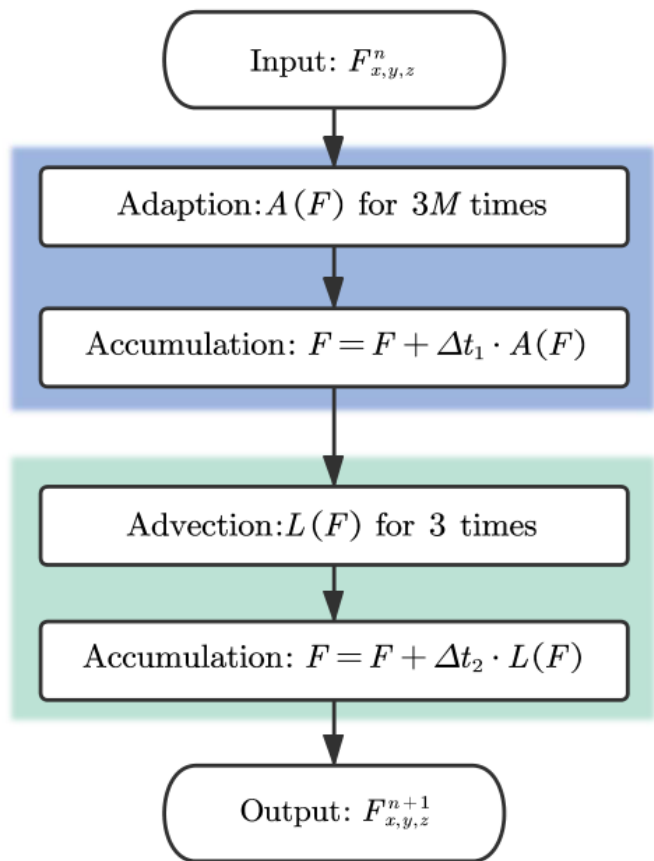
## 主要优化工作:

1. 三维并行剖分及优化
2. Z-prior性能优化
3. 高分辨率跳点差分优化
4. 耦合转换模块开发与改进
5. **通信与计算重构优化**

# 03 模式优化

## ● 通信合并与计算-通信重叠

- 分析大气模式动力框架的平流过程 (Advection) 与适应过程 (Adaption) 中的不同差分格式, 分别进行点到点通信的合并操作, 减少通信次数, 最大化利用通信带宽;
- 使用计算与通信的重叠 (overlap), 隐藏部分通信开销。



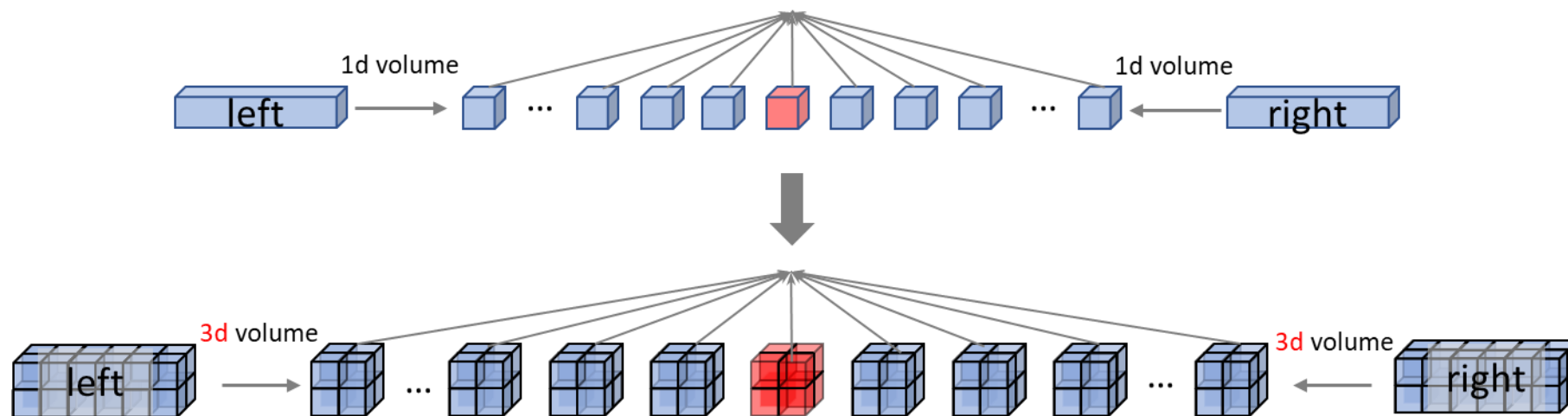
Process	Original difference terms	Leap-format difference gitterms	Variables
Adaption	$(x + 1, x)$	$(x + N_{leap}, x - N_{leap} + 1)$	$PXW, UT$
	$(x, x - 1)$	$(x + N_{leap} - 1, x - N_{leap})$	$PT, Pstar1, Pstar2, TT, \text{deltap}, GHI$
	$(x + 1, x - 1)$	$(x + 2 * N_{leap} - 1, x - 2 * N_{leap} + 1)$	$Pstar2$
Advection	$(x + 1, x)$	$(x + N_{leap}, x - N_{leap} + 1)$	$Ustar$
	$(x, x - 1)$	$(x + N_{leap} - 1, x - N_{leap})$	$Ustar$
	$(x + 1, x - 1)$	$(x + 2 * N_{leap} - 1, x - 2 * N_{leap} + 1)$	$UT, VT, TT$



# 03 模式优化

## ● 滤波平滑模块的计算重构优化:

- 大循环分解;
- 减少通信次数;
- 三维变量合并;
- 高维滤波迭代统一



# 03 模式优化

## ● 集合通信优化:

- 优化代码逻辑结构, 将开销较大的MPI全收集调用替换为MPI规约、前缀规约等操作;
- 使用非阻塞的集合通信;

算法 3:  $\sigma$ -表面垂直速度计算原始并行算法.

```

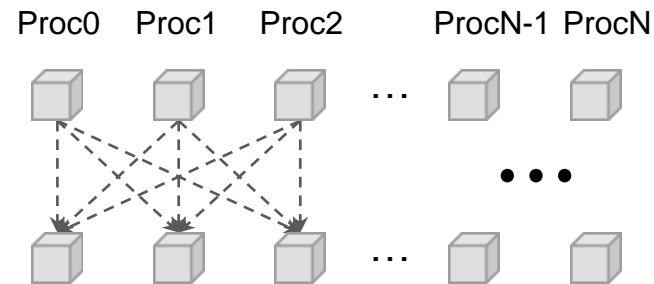
// Divergence of (PT*VT), Coefficients, Number of vertical layers
输入:  $\delta, coef, L$ 
输出:  $\sigma$  // Sigma-surface vertical velocity
1 do in parallel
  // Allgather along Z
2  MPI_Allgather( $\delta, \dots, \delta^L, \dots$ );
3  for  $k \leftarrow 1$  to  $L$  do
4     $\epsilon \leftarrow \epsilon + \delta^L(k) * coef(k)$ ; // Vertical sum
5  for  $k \leftarrow 1$  to  $L - 1$  do
6     $\delta^L(k) \leftarrow \delta^L(k) + \epsilon$ ;
7  for  $k \leftarrow 2$  to  $L$  do
8     $\sigma^L(k) \leftarrow \sigma^L(k - 1) + \delta^L(k)$ ; // Prefix sum
9  for  $k \leftarrow beg$  to  $end + 1$  do
10    $\sigma(k) \leftarrow \sigma^L(k)$ ;
  
```

算法 4:  $\sigma$ -表面垂直速度重构计算的并行算法.

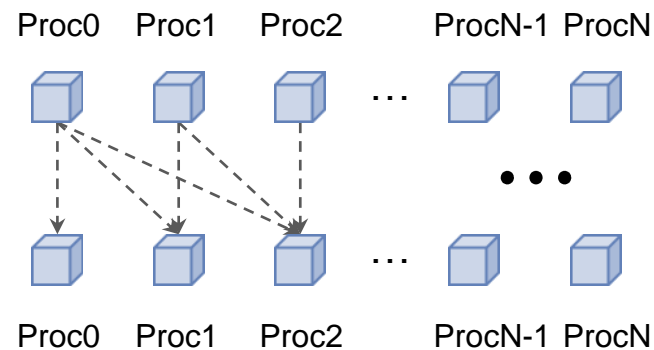
```

// Divergence of (PT*VT), Coefficients
输入:  $\delta, coef$ 
输出:  $\sigma$  // Sigma-surface vertical velocity
1 do in parallel
  // Local sum/prefix sum along Z
2  for  $k \leftarrow beg$  to  $end$  do
3     $\sigma(k + 1) \leftarrow \sigma(k) + \delta(k) * coef(k)$ ;
4     $\epsilon \leftarrow \epsilon + \delta(k) * coef(k)$ ;
  // Allreduce(Sum) along Z
5  MPI_Allreduce( $\epsilon, \dots, \epsilon^{sum}, \dots, sum, \dots$ );
  // Scan(Prefix sum) along Z
6  MPI_Exscan( $\epsilon, \dots, \epsilon^{scan}, \dots, sum, \dots$ );
7  for  $k \leftarrow beg$  to  $end + 1$  do
8     $\sigma(k) \leftarrow \sigma(k) + \epsilon^{sum} + \epsilon^{scan}$ ;
  
```

### Allgather



### Scan



01

分系统介绍

---

02

程序代码结构

---

03

模式优化

---

04

新建case、编译、运行

---



# 04 模式编译运行

## 1. 新建case:

- 进入到模式根目录下的/scripts目录, 通过命令 `./ create_newcase -case .././cas_esm_dev_rundir/case/[case目录名] -compset [配置名] -res [分辨率] -mach [集群配置名]` 创建算例;
- 进入 `/public/home/caohang/cas_esm_dev_rundir/case/[case目录名]` 下, 修改 `env_conf.xml`、`env_build.xml`、`env_mach_pes` 等文件;
- 通过 `./configure -case` 完成新建case

```

1.9x2.5_gx1v6 B_1850_WACCM_CN
1.9x2.5_gx1v6 B_1850-2005_WACCM_CN
1.9x2.5_gx1v6 B_1955-2005_WACCM_CN
1.9x2.5_gx1v6 B_RCP2.6_WACCM_CN
1.9x2.5_gx1v6 B_RCP4.5_WACCM_CN
1.9x2.5_gx1v6 B_RCP8.5_WACCM_CN

T31_gx3v7 B_1850_CN
T31_gx3v7 B_1850-2000_CN

If the user is interested in running a "stand-alone" component configuration,
the following model configurations have been validated scientifically and
have associated diagnostic output as part of the release:

1.9x2.5_1.9x2.5 F_2000_WACCM
1.9x2.5_1.9x2.5 F_AMIP_CAM5
0.9x1.25_0.9x1.25 F_AMIP_CAM5
1.9x2.5_1.9x2.5 F_AMIP_CN
0.9x1.25_0.9x1.25 F_AMIP_CN

0.9x1.25_gx1v6 I_2000
0.9x1.25_gx1v6 I_2000_CN

T62_gx1v6 C_NORMAL_YEAR

For more information regarding alternative component configurations,
please refer to the individual component web pages at
http://www.cesm.ucar.edu/models/cesm1.0

*****
Component set F_2000_CAM5_nochem (FC5NCH)
Desc Stand-alone cam default, prescribed ocn/ice, cam5 physics
*****

Creating /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH
Locking file /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH/env_case.xml
Successfully created the case for sugon_ch

```

```

./configure -case
./CASE_MACHINE.build

<entry id="NTASKS_ATM" value="1024" />
<entry id="NTHRDS_ATM" value="1" />
<entry id="ROOTPE_ATM" value="0" />

<entry id="NTASKS_WRF" value="1024" />
<entry id="NTHRDS_WRF" value="1" />
<entry id="ROOTPE_WRF" value="0" />

<entry id="NTASKS_GEA" value="1024" />
<entry id="NTHRDS_GEA" value="1" />
<entry id="ROOTPE_GEA" value="0" />

<entry id="NTASKS_SRD" value="1024" />
<entry id="NTHRDS_SRD" value="1" />
<entry id="ROOTPE_SRD" value="0" />

<entry id="NTASKS_LND" value="1024" />
<entry id="NTHRDS_LND" value="1" />
<entry id="ROOTPE_LND" value="0" />

<entry id="NTASKS_ICE" value="120" />
<entry id="NTHRDS_ICE" value="1" />
<entry id="ROOTPE_ICE" value="0" />

<entry id="NTASKS_OCN" value="120" />
<entry id="NTHRDS_OCN" value="1" />
<entry id="ROOTPE_OCN" value="0" />

<entry id="NTASKS_CPL" value="1024" />
<entry id="NTHRDS_CPL" value="1" />
<entry id="ROOTPE_CPL" value="0" />

<entry id="NTASKS_GLC" value="1024" />
<entry id="NTHRDS_GLC" value="1" />
<entry id="ROOTPE_GLC" value="0" />

```

```

adding use_case 2000_control defaults for var sim_year with val 2000
adding use_case 2000_control defaults for var sim_year_range with val constant
adding use_case 2000_control defaults for var use_case_desc with val Conditions to simulate 2000 land-use
cat
cat lnd_in
cat
if ( docn == licom ) then
rm -f /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH/Buildconf/clmconf/lnd_in /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH/Buildconf/clmconf/cesm_nameлист
cat
cat /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH/Buildconf/clmconf/Filepath
cat /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH/Buildconf/clmconf/CESM_cppdefs
cat
rm -f /work/zhanghe/caohang/cas_esm_dev_rundir/case/FFD14C5NCH/Buildconf/clmconf/Filepath
configure done.
Successfully generated resolved namelist, prestage, and build scripts
Locking file env_conf.xml
Generating clean_build script
Generating submit script
Generating build script
Generating run script
Locking file env_mach_pes.xml
Successfully configured the case for sugon_ch
If an old file exists for this case, you might want to
run the *.clean_build script before building

```

# 04 模式编译运行

## 2. 编译case:

- 在/public/home/caohang/cas\_esm\_dev\_rundir/case/[case目录名]文件夹下, 输入命令./\*.run或者./recompile.sh进行程序编译;

```
[caohang@login01 FFD14C5NCH]$ ./recompile.sh
/public/home/caohang/cas_esm_dev_rundir/case/FFD14C5NCH
./FFD14C5NCH.sugon_ch.build

-----
CCSM BUILDNML SCRIPT STARTING
- To prestage restarts, untar a restart.tar file into /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run
CCSM BUILDNML SCRIPT HAS FINISHED SUCCESSFULLY
-----

CCSM PRESTAGE SCRIPT STARTING
- CCSM input data directory, DIN_LOC_ROOT_CSMDATA, is /public/home/haohq/inputdata
- Case input data directory, DIN_LOC_ROOT, is /public/home/haohq/inputdata
- Checking the existence of input datasets in DIN_LOC_ROOT
CCSM PRESTAGE SCRIPT HAS FINISHED SUCCESSFULLY
-----

CCSM BUILDEXE SCRIPT STARTING
- Build Libraries: mct pio csm_share
Thu Jun 10 19:53:13 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/mct/mct.bldlog.210610-195311
Thu Jun 10 19:53:14 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/pio/pio.bldlog.210610-195311
Thu Jun 10 19:53:16 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/csm_share/csm_share.bldlog.210610-195311
Thu Jun 10 19:53:17 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/cpl.bldlog.210610-195311
Thu Jun 10 19:53:17 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/atm.bldlog.210610-195311
Thu Jun 10 19:53:24 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/wrf.bldlog.210610-195311
Thu Jun 10 19:53:24 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/gea.bldlog.210610-195311
Thu Jun 10 19:53:24 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/lnd.bldlog.210610-195311
Thu Jun 10 19:53:25 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/ice.bldlog.210610-195311
Thu Jun 10 19:53:26 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/ocn.bldlog.210610-195311
Thu Jun 10 19:53:26 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/glc.bldlog.210610-195311
Thu Jun 10 19:53:27 CST 2021 /public/home/caohang/cas_esm_dev_rundir/FFD14C5NCH/run/ccsm.bldlog.210610-195311
- Locking file env_build.xml
- Locking file Macros.sugon_ch
CCSM BUILDEXE SCRIPT HAS FINISHED SUCCESSFULLY
```



# 04 模式编译运行

## 3. 模式运行:

- 输入文件与参数: 进入到/public/home/caohang/cas\_esm\_dev\_rundir/[作业执行目录]下, 设置atm\_in和drv\_in中的进程数、进程剖分、输入文件等配置;
- 作业提交脚本: 使用./run.sh命令执行作业脚本slurm.job的提交运行。slurm.job设置如下

```
&solar_inparm
solar_const      = 1361.27
solar_htng_spctrl_scl = .true.
/
&spmd_iap_inparm
dtdy             = 200.000
ndq              = 3
ndt              = 3
npr_yz          = 64,16,16,64
zmh_cfl_drag     = .true.
zmh_cfl_u0r     = 0.8000
decomp_flag     = .false.
/
&tropopause_nl
tropopause_climo_file = '/public/
/
&uwshcu_nl
uwshcu_rpen     = 10.0
/
&zmconv_nl
zmconv_c0_lnd   = 0.005900
zmconv_c0_ocn   = 0.045000
zmconv_dmpdz    = -1.0E-3
zmconv_ke       = 1.0E-6
/
```

```
&seq_timemgr_inparm
calendar        = 'NO_LEAP'
atm_cpl_dt      = 1800
lnd_cpl_dt      = 1800
ocn_cpl_dt      = 1800
ice_cpl_dt      = 1800
glc_cpl_dt      = 86400
start_ymd       = 00010101
start_tod       = 0
stop_option     = 'ndays'
stop_n          = 3
stop_ymd        = -999
restart_option  = 'ndays'
restart_n       = 5
restart_ymd     = -999
end_restart    = .false.
history_option  = 'never'
history_n       = -999
history_ymd     = -999
histavg_option  = 'never'
histavg_n       = -999
histavg_ymd    = -999
tprof_option    = 'never'
tprof_n         = -999
tprof_ymd      = -999
/
```

```
&ccsm_pes
atm_ntasks      = 1024
atm_nthreads    = 1
atm_rootpe      = 0
atm_pestride    = 1

lnd_ntasks      = 1024
lnd_nthreads    = 1
lnd_rootpe      = 0
lnd_pestride    = 1

ice_ntasks      = 120
ice_nthreads    = 1
ice_rootpe      = 0
ice_pestride    = 1

ocn_ntasks      = 120
ocn_nthreads    = 1
ocn_rootpe      = 0
ocn_pestride    = 1

glc_ntasks      = 1024
glc_nthreads    = 1
glc_rootpe      = 0
glc_pestride    = 1

cpl_ntasks      = 1024
cpl_nthreads    = 1
cpl_rootpe      = 0
cpl_pestride    = 1
```

```
#!/bin/bash
#SBATCH -J FFD14
#SBATCH -p normal
##SBATCH -p dcu
##SBATCH -p large
#SBATCH --mem=200G
#SBATCH -N 32
#SBATCH -n 1024
#SBATCH --ntasks-per-node=32 # 64 cores max in each node
#SBATCH --ntasks-per-socket=4 # 8 cores max in each socket 8 sockets
#SBATCH --cpus-per-task=2
#SBATCH -o %j.log
##SBATCH -e log.%j
#SBATCH --exclusive
#SBATCH -t 30:00
#SBATCH -x b2101r2n1

module purge
module load compiler/intel/2017.5.239
#module load mpi/intelmpi/2017.4.239
module load mathlib/netcdf/intel/4.4.1
module load mpi/hpcx/2.7.4/intel-2017.5.239
module li

ulimit -s unlimited

scontrol show hostname > nd
NP=$SLURM_NPROCS
mpirun -np $NP ./ccsm.exe
#srn ./ccsm.exe
```

## 3. 模式运行:

- 输入文件与参数: 进入到/public/home/caohang/cas\_esm\_dev\_rundir/[作业执行目录]下, 设置atm\_in和drv\_in中的进程数、进程剖分、输入文件等配置;
- 作业提交脚本: 使用./run.sh命令执行作业脚本slurm.job的提交运行。slurm.job设置如下
- 查看作业执行log (SYPD等)

```

***** END OF MODEL RUN *****

(seq_mct_drv): ===== SUCCESSFUL TERMINATION OF CPL7-CCSM =====
(seq_mct_drv): ===== at YMD,TOD = 10106 0 =====
(seq_mct_drv): ===== # simulated days (this run) = 5.000 =====
(seq_mct_drv): ===== compute time (hrs) = 0.013 =====
(seq_mct_drv): ===== # simulated years / cmp-day = 24.552 =====
(seq_mct_drv): ===== pes min memory highwater (MB) 176.259 =====
(seq_mct_drv): ===== pes max memory highwater (MB) 359.885 =====
(seq_mct_drv): ===== pes min memory last usage (MB) 674.456 =====
(seq_mct_drv): ===== pes max memory last usage (MB) 1033.005 =====
  
```

谢 谢