



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

培训人：韦俊林

2022年6月1日

01

分系统介绍

02

程序代码结构

03

模式优化

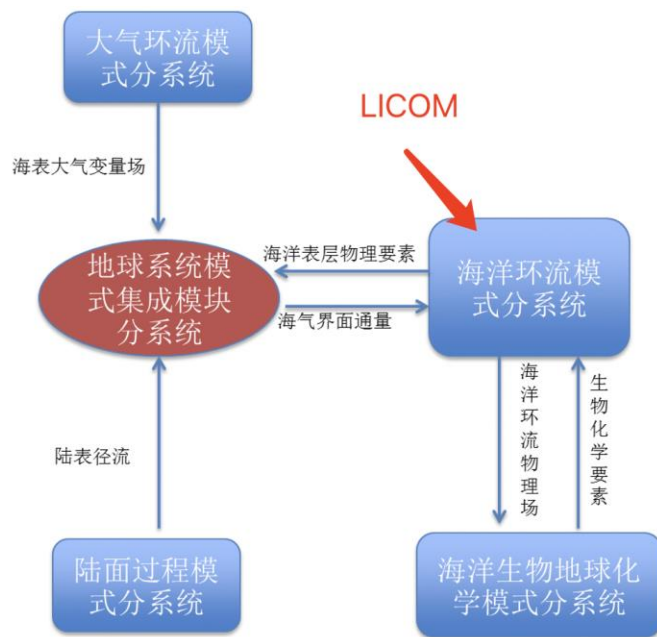
04

新建case、编译、运行

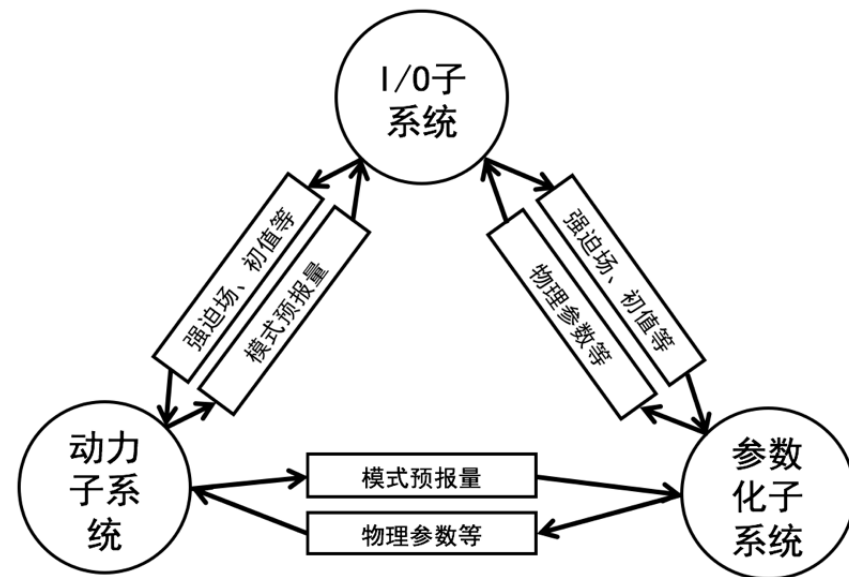
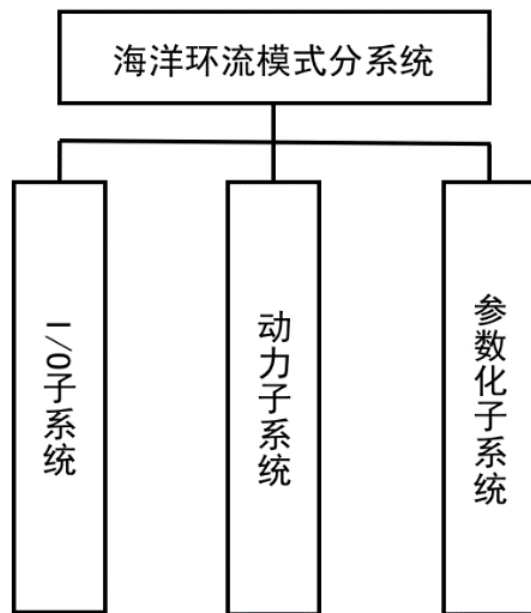


# 分系统介绍

- 海洋环流模式(LICOM)外部耦合
  - 大气环流模式分系统
  - 陆面模式分系统
  - 海洋生物化学分系统
- 海洋环流模式包括
  - CPU版本
  - DCU版本



- 海洋环流模式组成
  - 海洋模式
  - 海冰模式
- 海洋环流模式包括三个子系统
  - I/O子系统
  - 动力子系统
  - 参数化子系统



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

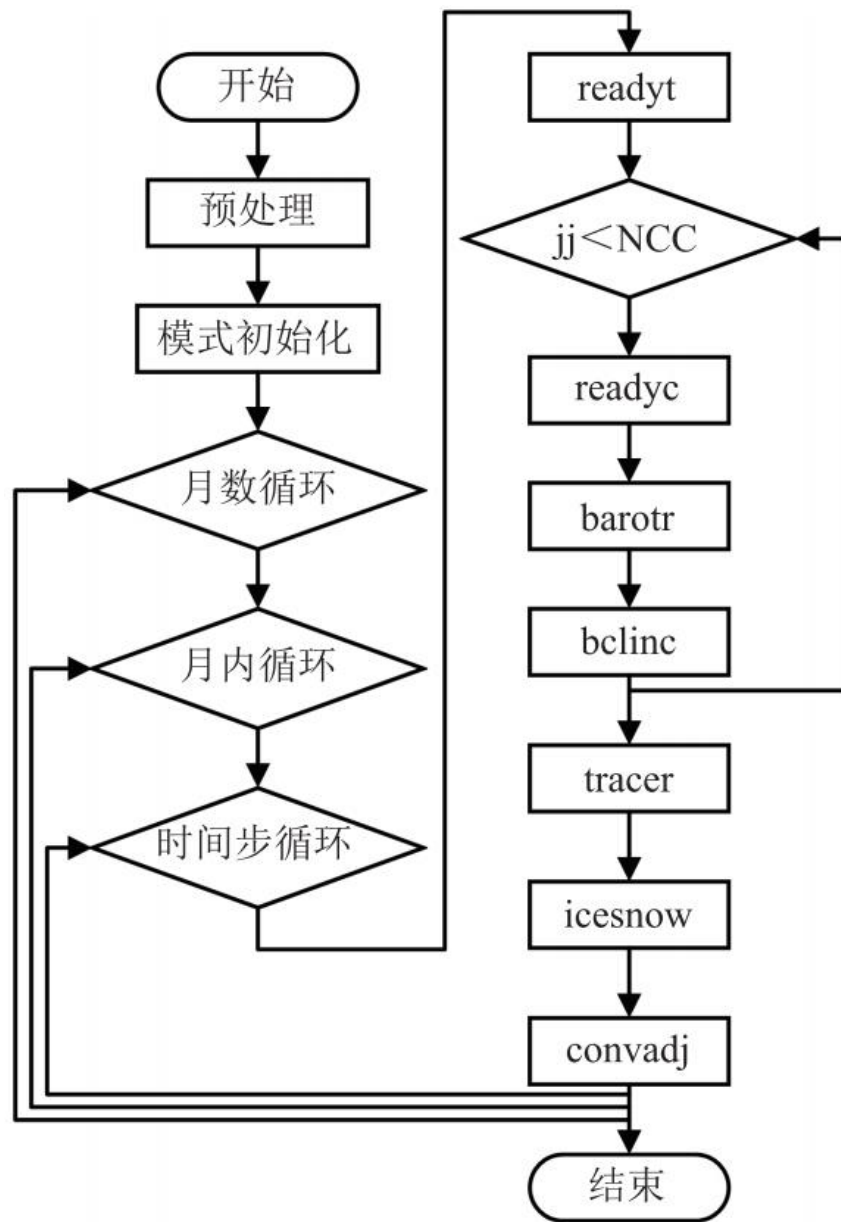
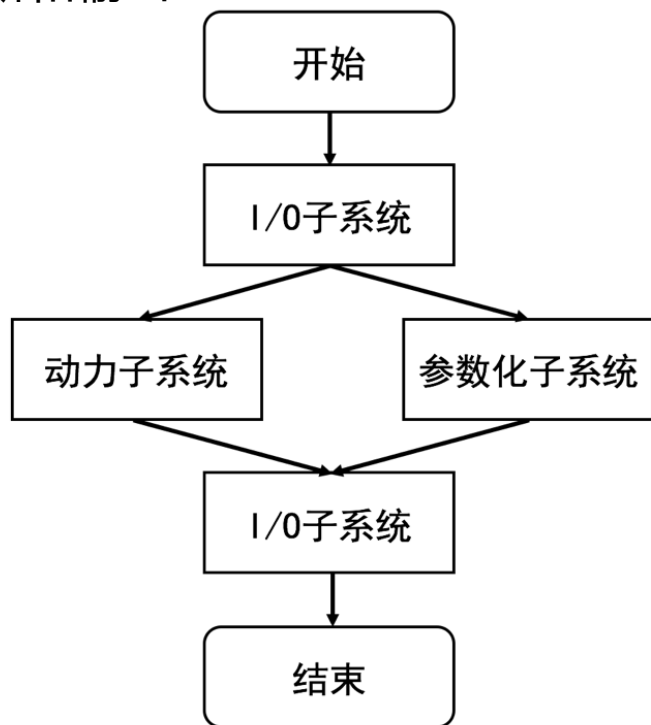
---



# 程序流程介绍

## 海洋环流模式运行流程:

- 预处理
- 模式初始化
- 模式迭代计算
  - 强迫场更新
  - 核心迭代
  - 每日诊断和输出
- 模式结束



# 程序文件结构

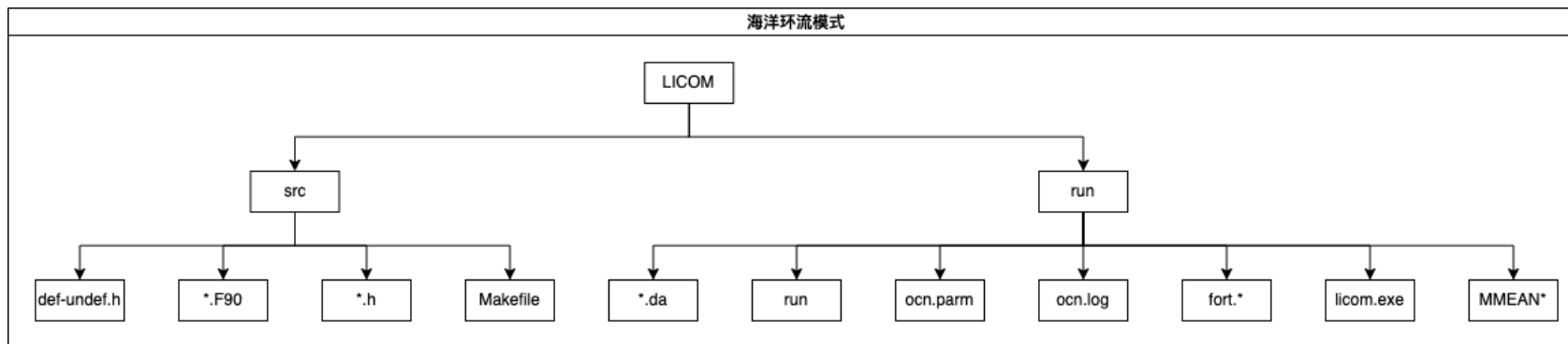
- 源码目录 (src)
  - Makefile
  - def-undef.h
  - \*.F90
  - Clockof.c
  - \*.h
- 运行目录 (run)
  - licom.exe
  - ocn.parm
  - \*.da
  - \*.nc
  - fort.\*
  - MMEAN\*
  - ...

定义项目编译规则  
宏定义/功能选择开关  
源码  
C语言计时函数  
头文件

LICOM可执行文件  
设置程序的参数  
海洋/网格信息  
强迫场  
每日输出  
月平均输出

源码目录示意图

```
accumm_daily_avg.F90  diag_mod.F90          licomcpl7.F90         readyc.F90
accumm_daily.F90      distribution.F90       LICOM_Error_mod.F90  readyt.F90
accumm.F90            domain.F90            mm00.F90             run_time.F90
addps.F90            dyn_mod.F90          msg_mod.F90         smag.F90
advection.F90        energy.F90           nextstep.F90        smuvh.F90
barotr.F90           flux_cpl.F90         operators.F90        ssave-cdf-budgetdailyavg.F90
bc1inc.F90          forc_mod.F90         output_1dto4d.F90   ssave-cdf-budget.F90
blocks.F90          gather_scatter.F90   output_mod.F90      ssave-cdf-dailyavg.F90
boundary.F90        global_reductions.F90 param_mod.F90       ssave-cdf-day.F90
broadcast.F90       grid.F90             pconst_mod.F90     ssave-cdf-instant.F90
buf_mod.F90         grids.F90            pmix_mod.F90        ssave-cdf-instant-step.F90
canuto_2002.F90     hmix_del2.F90        POP_BlocksMod.F90  ssave-cdf-month.F90
canuto_2010_mod.F90 hmix_del4.F90        POP_CommMod.F90    ssave-cdf-month-omip.F90
canuto_ini_2002.F90 icesnow.F90         POP_CplIndices.F90 ssave-cdf-testinput.F90
canuto_mod_2002.F90 inirun.F90          POP_DistributionMod.F90 sw_absor.F90
cdf_mod.F90         invtri.F90          POP_FieldMod.F90   sw_mod.F90
clockof.c           isoadv1.F90.save    POP_GridHorzMod.F90 thermalexpansion.F90
constant_mod.F90    isoadv.F90          POP_HaloMod.F90    tracer.F90
control_mod.F90     isoflux.F90         POP_ReductionsMod.F90 tracer_mod.F90
convadj.F90        isopyc.F90          POP_SpaceCurveMod.F90 upwell.F90
core_daily.F90     isopyc_mod.F90     post_cpl.F90       utils.F90
dailyavgini.F90    isopyi.F90         precision_mod.F90  vinteg.F90
def-undef.h        k1_3.F90            prepare_output.F90 work_mod.F90
dens.F90           k2_3.F90            rdriver.F90        wrap_nf.F90
density.F90        k3_123.F90         read_tsforres.F90  yy00.F90
```



### 主要预编译选项

名称	功能	备注
SPMD	MPI并行的开关	当前版本必须打开
SYNCH	同步积分的开关	
FRC_ANN	年平均气候强迫的开关	
CDF_IN	采用NetCDF格式输入的开关	
FRC_DAILY	逐日强迫开关	耦合框架下运行时用不到
FRC_CORE	CORE强迫开关	耦合框架下运行时用不到
BIHAR	双调和水平混合方案的开关	
COUP	在耦合框架下运行开关	
SOLAR	采用太阳短波辐射穿透参数化的开关	
HIGHRES/LOWRES	分辨率开关	只能选其一
TSPAS	采用两步正定保形平流方案的开关	
ISO	中尺度涡参数化 (GM90) 的开关	
SOLARCHLORO	考虑叶绿素影响的短波穿透参数化开关	
SMAG	Smagorinsky水平混合方案的开关	
TIDEMIX	潮汐混合开关	
BCKMEX	内波混合开关	
LDD97	中尺度涡参数化LDD97方案开关	
ISO_TYPE_BF	中尺度涡参数化BF型方案开关	
SSSNORM	表面盐度回复开关	
NODIAG	将温盐过程关闭, 该选项主要用于模式调试	

### 并行计算网格划分参数

名称	功能
NIMT	水平方向网格点数
NJMT	
NKM	垂直层数
BLCKX	对应于NIMT方向的网格块划分
BLCKY	对应于NJMT方向的网格块划分

### 网格剖分算法:

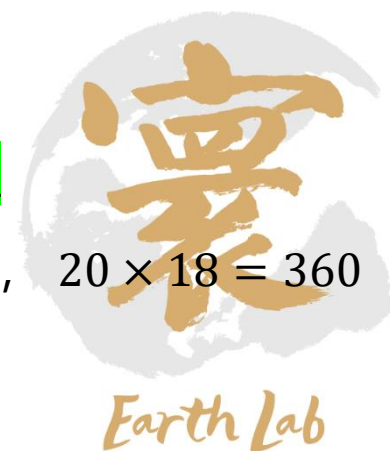
- 假设共使用N个进程, IMT和JMT方向分别分配x和y个进程,  $x \times y = N$ , 则

$$\begin{cases} BLCKX = \lceil NIMT/x \rceil \\ BLCKY = \lceil NJMT/y \rceil \end{cases}$$

- 例: 10km分辨率, 使用360进程

$$\begin{cases} NIMT = 3600 \\ NJMT = 2302 \end{cases}, \begin{cases} x = 20 \\ y = 18 \end{cases}, 20 \times 18 = 360$$

$$\begin{cases} BLCKX = \lceil 3600/20 \rceil = 180 \\ BLCKY = \lceil 2302/18 \rceil = 128 \end{cases}$$





# 程序输入文件说明

## 运行 (run) 目录

### ocn.parm 部分参数

```

out_dir = "./"
adv_momentum = "centered"
adv_tracer = "tspas"
boundary_restore = 2 ! 0 for no restoring
/
&domain_nml
  nprocs_clinic = 40  进程数
  nprocs_tropic = 40
  clinic_distribution_type = 'balanced'
  tropic_distribution_type = 'cartesian'
  ew_boundary_type = 'cyclic'
  ns_boundary_type = 'tripole'
  profile_barrier = .false.
/
&grid_nml  网格namelist
  horiz_grid_file = 'grid.da'
  vert_grid_file = 'depth.da'
  topography_file = 'kmt.da'
  basin_grid_file = 'Basin.data'
  horiz_grid_opt = 'tripole'
/
    
```



输入文件	内容
ocn.parm	输入参数的namelist
grid.da	海洋水平网络文件
kmt.da	海洋垂直网格层文件
depth.da	模式层深度
basin.da	海洋洋盆文件
sss_phc3_monthly.nc	海表面盐度恢复场
tidal_energy.nc	内潮能量通量文件
MODEL_CHLFR	叶绿素浓度文件
TSinitial	海洋模式观测初值场
salt_initial	Mecator场数据
ice_tripole.grid	海冰水平网格文件
ice_tripole.kmt	海冰垂直网格层文件
domain_licom.nc	海洋模式网格参数文件
domain.lnd.T62_T62.090129.nc	陆地网格参数文件
seq_maps.rc	各模式分量网格插值定义文件
camT62_to_licomtripole_aave_frac_20160507.nc	大气向海洋模式网格匹配插值文件
licomtripole_to_camT62_aave_frac_20160507.nc	海洋向大气模式网格匹配插值文件
runoff1x1_to_licomtripole_aave_frac_20160507.nc	陆地径流分量向海洋模式网格匹配插值文件
...	...

强迫场输入文件	
输入文件	内容
t_10.2016_daily.nc	10m大气温度强迫场定义文件
u_10.2016_daily.nc	10m纬向风强迫场定义文件
v_10.2016_daily.nc	10m经向风强迫场定义文件
q_10.2016_daily.nc	10m大气比湿强迫场定义文件
slp.2016_daily.nc	海表面气压强迫场定义文件
rsds.2016_daily.nc	向下的短波辐射强迫场定义文件
rlsds.2016_daily.nc	向下的长波辐射强迫场定义文件
rain.2016_daily.nc	降雨量（仅海面）强迫场定义文件
snow.2016_daily.nc	降雪量强迫场定义文件
runoff_all.2016_daily.nc	陆地径流强迫场定义文件
ice.2016_daily.nc	冰面积分数强迫场定义文件

```

[esm7@login09 100km]$ ls -hts
total 4.2G
2.6K cp-100km-data.sh
95M fort.22.0002-01-01
512 rpointer.ocn
1.5K ocn.parm
286M q_10.2016_daily.nc
14M chl_clm_monthly_tripole_360X218X30_omp_db_20160507.nc
512 depth.da
4.2M grid.da
614K kmt.da
4.2M LICOM_Hor_Grid_360X218X30_tripole_omp_20160507.DATA
286M slp_2016_daily.nc
14M sss_month_phc3_tripole_360X218X30_omp_db_20160507.nc
6.6M tidal_energy_licom_tripole_360X218X30_omp_db_20160507.nc
9.0K dncoef_eq1x1.h
286M rain_2016_daily.nc
307K LICOM_Basin_360X218X30_tripole_omp_20160507.DATA
614K LICOM_Topo_360X218X30_tripole_omp_20160507.DATA
esm7@login09 100km$
    
```

```

[esm7@login09 10km]$ ls -hts
total 29G
1.5K cp-10km-data.sh
512 rpointer.ocn
18G fort.22.0002-01-05
1.5K ocn.parm
1.4G chl_clm_monthly_tripole_3600X2302X55_omp_db_20170529.nc
15K dncoef_h1
1.4G sss_month_phc3_tripole_3600X2302X55_omp_db_20170529.nc
696M tidal_energy_licom_tripole_3600X2302X55_omp_db_20170529.nc
1.8G mecaton-temp-20160501-3600X2302X55_n2s.nc
1.8G mecaton-salt-20160501-3600X2302X55_n2s.nc
286M prec_2016_daily.nc
286M u_10.2016_daily.nc
286M v_10.2016_daily.nc
286M t_10.2016_daily.nc
286M u_10.2016_daily.nc
286M snow_2016_daily.nc
286M runoff_all_2016_daily.nc
286M slp_2016_daily.nc
286M rsds_2016_daily.nc
286M rlds_2016_daily.nc
286M rain_2016_daily.nc
286M ice_2016_daily.nc
286M q_10.2016_daily.nc
286M rain_2016_daily.nc
286M prec_2016_daily.nc
286M dn10.2016_daily.nc
64M kmt.da
512 depth.da
443M grid.da
    
```

100km输入数据

10km算例



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

---



# 模式优化加速

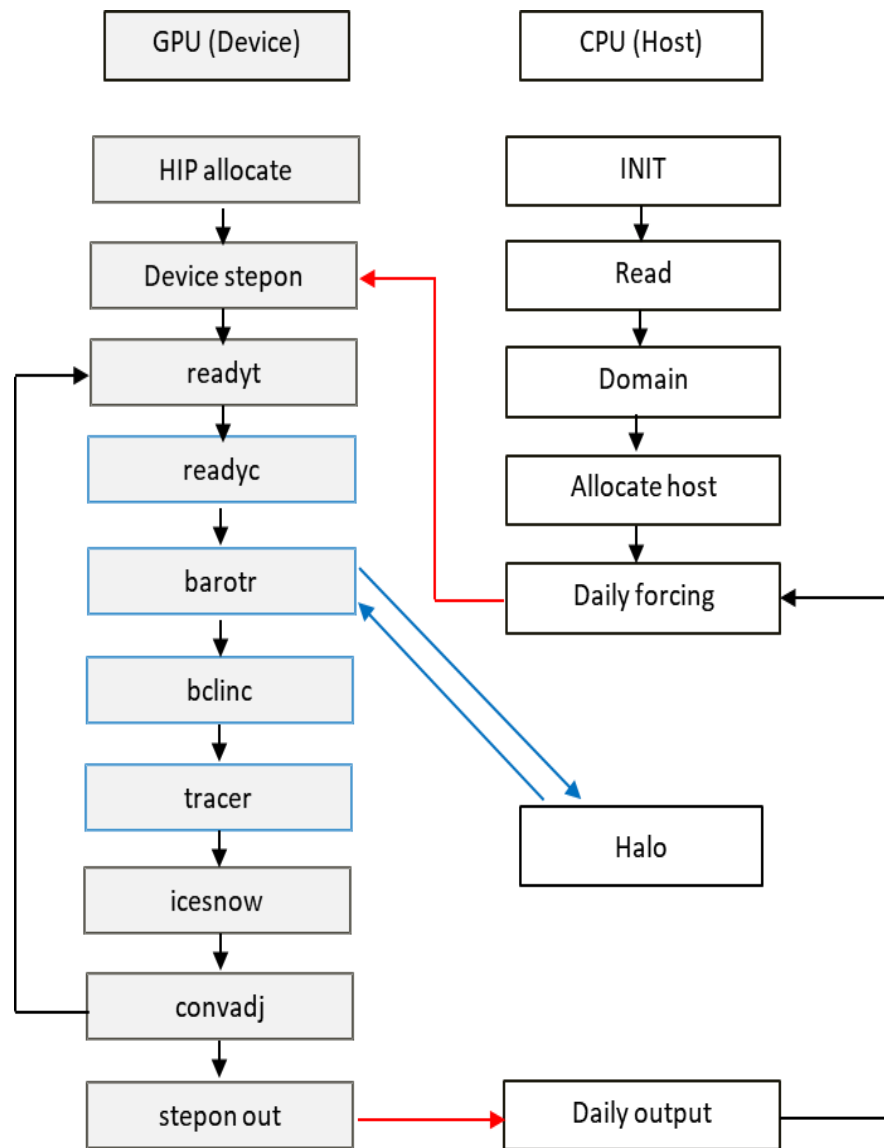
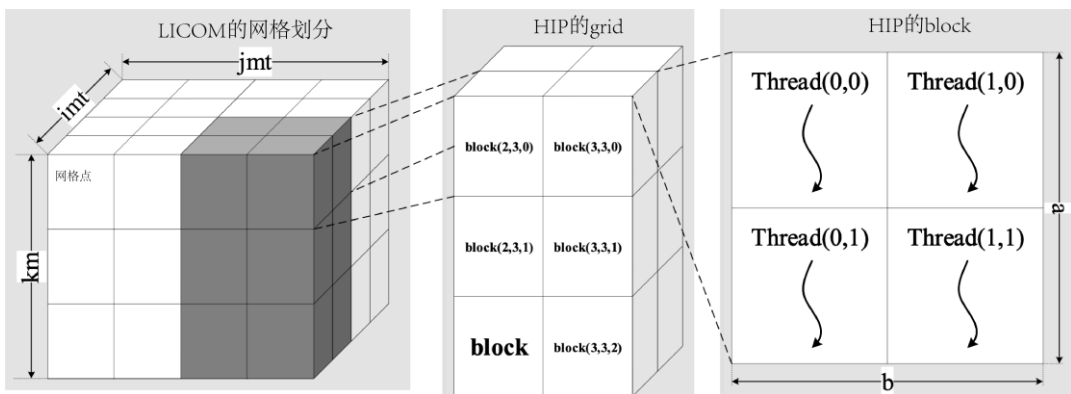
- 海洋环流模式的DCU优化加速

F90文件109个

C++(HIP)文件12个, C文件2个

h文件32个

- 共约63 k行代码(增加10 k行);



# 模式优化加速

## CPU优化

- **内存管理优化**: 部分数组改为动态数组, 解决高分辨率时CPU/DCU内存不够的问题
- **数据读取优化**: 增加读取量减少读取次数, 并采用并行读取。时间开销从200秒降到20秒以下。
- **数据输出优化**: 输出进行并行化优化, 同时引入输出与计算重叠。时间开销从5分钟下降到1分钟。

## DCU优化

- **程序结构优化**: 在模式初始化阶段完成所有DCU内存的申请与初始化, 最大化减少异构计算数据传输开销。
- **特定函数优化**:
  - 为适配SIMT加速卡特性, 优化复杂的逻辑判断
  - 改写判断流程, 消除相邻格点的数据依赖
  - 以及其他结合SIMT加速卡特性的常规优化方案

## 数据依赖初始版本

```
--global__ void kernel_1(double *d_v) {
    int i = blockDim.x * blockIdx.x + threadIdx.x;
    int j = blockDim.y * blockIdx.y + threadIdx.y;
    int k = blockDim.z * blockIdx.z + threadIdx.z;
    if (k < KM && j < JMT && i < IMT) {
        d_v[k * JMT * IMT + j * IMT + i] = foo_1(i, j, k);
    }
    return ;
}

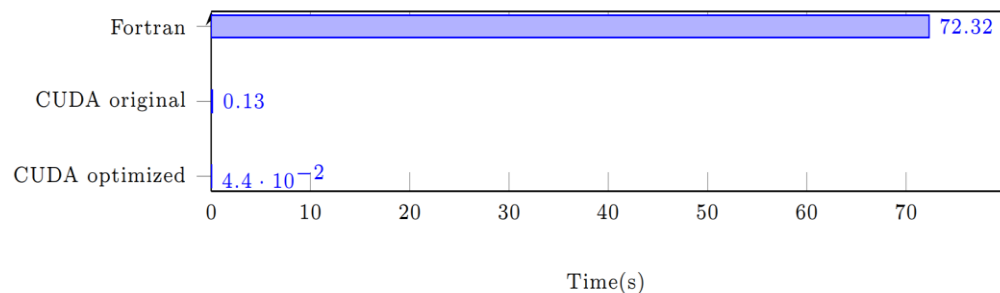
--global__ void kernel_2(double *d_u, double *d_v) {
    int i = blockDim.x * blockIdx.x + threadIdx.x;
    int j = blockDim.y * blockIdx.y + threadIdx.y;
    int k = blockDim.z * blockIdx.z + threadIdx.z;
    if (k >= 1 && k <= (KM-2) && j < JMT && i < IMT) {
        d_u[k * JMT * IMT + j * IMT + i] = foo_2(d_v[(k-1) * JMT *
        IMT + j * IMT + i], d_v[k * JMT * IMT + j * IMT + i], d_v[(k+1)
        * JMT * IMT + j * IMT + i]);
    }
    return ;
}
```

## 数据依赖优化版本

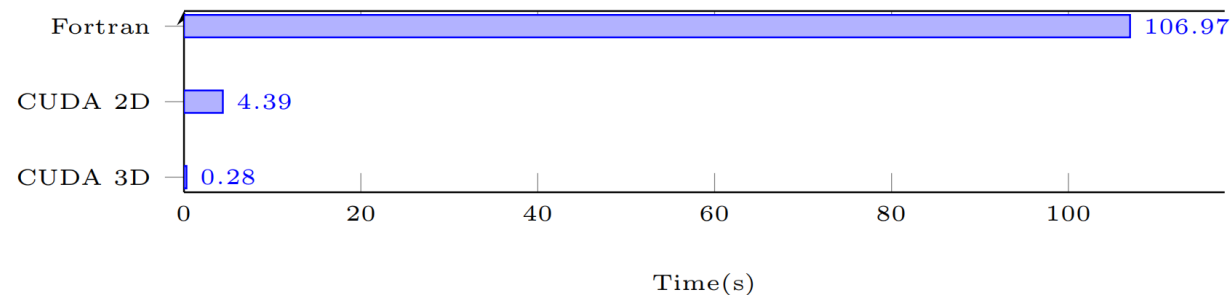
```
--global__ void kernel(double *d_u, double *d_v) {
    int i = blockDim.x * blockIdx.x + threadIdx.x;
    int j = blockDim.y * blockIdx.y + threadIdx.y;
    int k = blockDim.z * blockIdx.z + threadIdx.z;
    if (k < KM && j < JMT && i < IMT) {
        d_v[k * JMT * IMT + j * IMT + i] = foo_1(i, j, k);
    }
    if (k >= 1 && k <= (KM-2) && j < JMT && i < IMT) {
        double v_1 = foo_1(i, j, k-1);
        double v_2 = foo_1(i, j, k+1);
        d_u[k * JMT * IMT + j * IMT + i] = foo_2(v_1, d_v[k * JMT *
        IMT + j * IMT + i], v_2);
    }
    return ;
}
```

# 模式优化加速

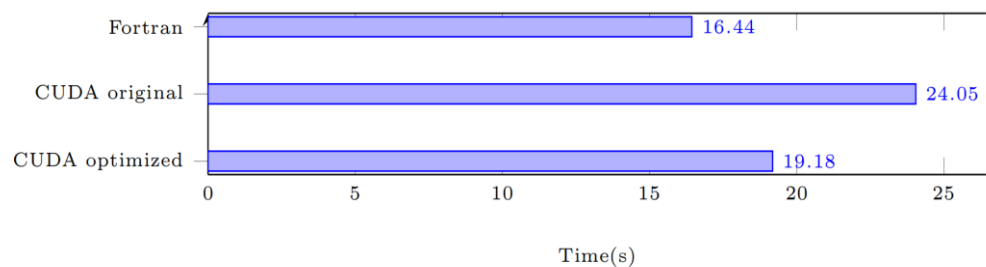
- 单节点，模块优化效果。
- 单位：时间，数值越小越好



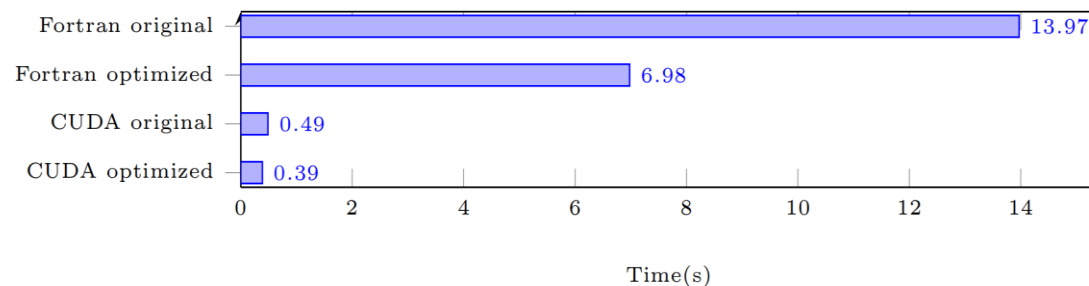
(a) Elapsed time of READYT



(b) Elapsed time of READYC



(c) Elapsed time of BAROTR

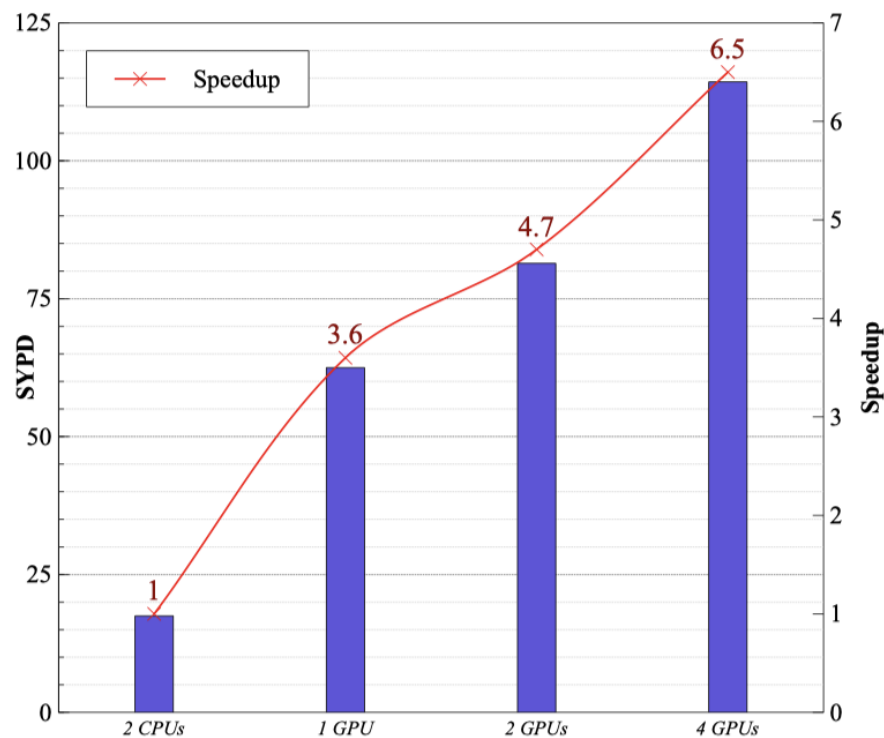
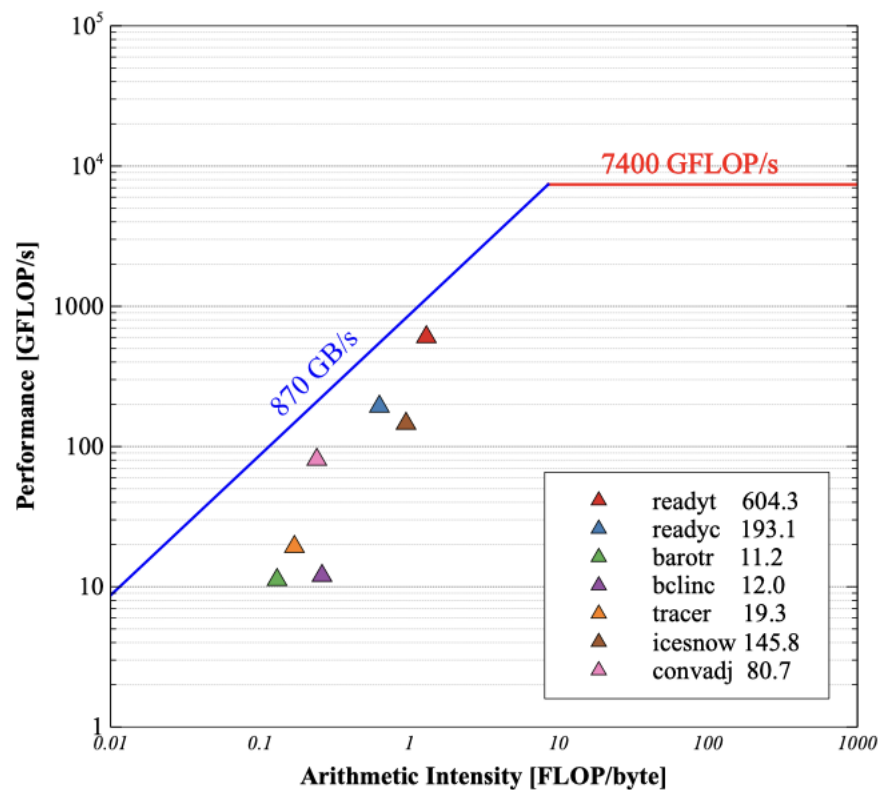


(d) Elapsed time of ADVECTION\_TRACER



# 模式优化加速

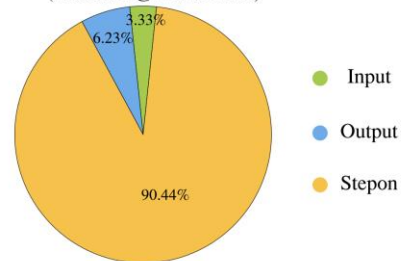
- 单节点测试
  - Roofline性能模型测试效果 (左)
  - 整体性能 (右)



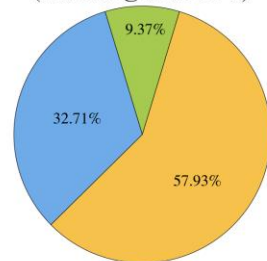
# 模式优化加速

- 大规模测试

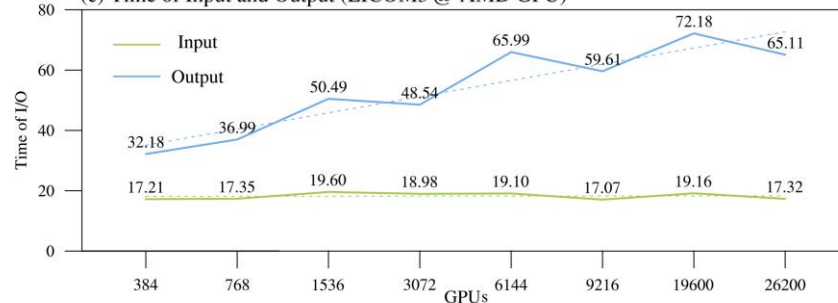
(a) The ratios of each parts at 384 GPUs (LICOM3 @ AMD GPU)



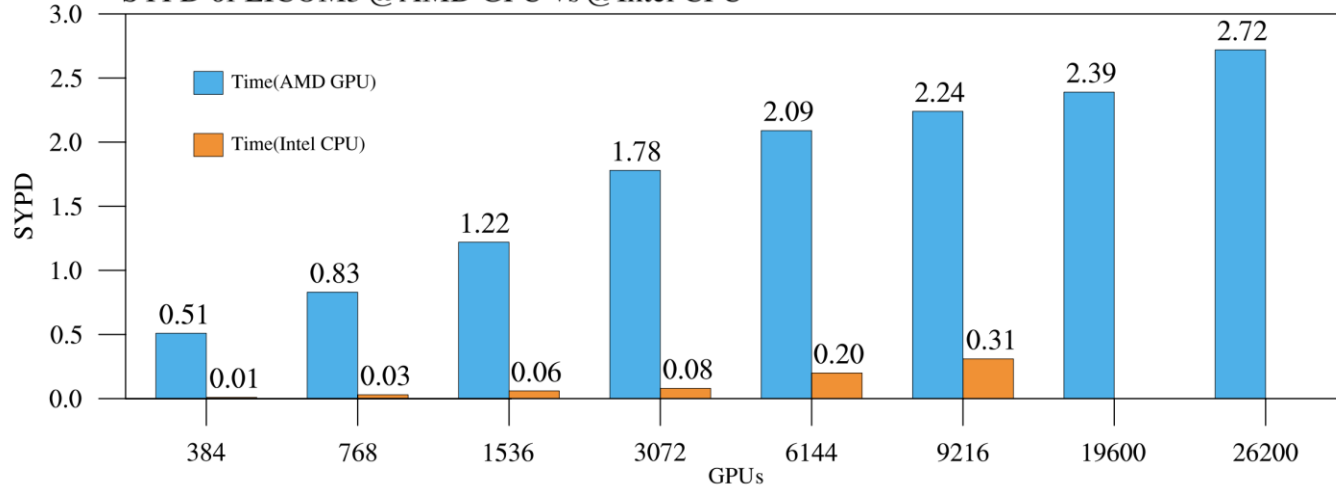
(b) The ratios of each parts at 9216 GPUs (LICOM3 @ AMD GPU)



(c) Time of Input and Output (LICOM3 @ AMD GPU)



SYPD of LICOM3 @AMD GPU vs @Intel CPU





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

---



# 新建case、编译、运行 (一)

- DCU版本海洋环流模式为例 (CPU版本海洋环流模式类似)
  - 第一步: load相应必要的软件包、库、编译环境等
    - module load compiler/intel/2017.5.239
    - module load mpi/hpcx/2.7.4/intel-2017.5.239
    - module load mathlib/netcdf/intel/4.4.1
    - module load compiler/rocm/3.3
  - 第二步: 在 src 目录下, 编辑 “def-undef.h” 文件 (见右图), 确定选用的功能, 设置网格划分方案

```
#define SPMD
#define SYNCH
#undef FRC_ANN
#define CDFIN
#undef FRC_DAILY
#define FRC_CORE
#define SOLAR
#define ACOS
#define BIHAR
#undef SMAG_FZ
#undef SMAG_OUT
#define NETCDF
#undef BOUNDARY
#define NODIAG
#undef ICE
#undef SHOW_TIME
#undef DEBUG
#undef COUP
#undef ISO
#define D_PRECISION
#define CANUTO
#undef SOLARCHLORO
#undef BCKMEX
#define TIDEMIX
#define SSSNORM
#undef LDD97
#undef SMAG
#define TEST_IDEAL
#undef LOWRES
#define HIGHRES
#undef SUPHIGH
#undef DAILYACC
#undef DAILYBUGDET
#define BLCKX 60
#define BLCKY 63
#define NJMT 2302
#define NIMT 3600
#define NKM 55
#define MXBLCKS 1
~
```



# 新建case、编译、运行 (二)

- DCU版本海洋环流模式为例

- 第三步**: 在 src 目录下, 输入 “make” 命令, 编译程序 (见下图)

```
[hydeng@login03 run-2220]$ ./build.sh
Currently Loaded Modulefiles:
  1) compiler/rocm/3.3                2) compiler/intel/2017.5.239        3) mpi/hpcx/2.7.4/intel-2017.5.239  4) mathlib/netcdf/intel/4.4.1
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c shr_kind_mod.F90 -o shr_kind_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c shr_log_mod.F90 -o shr_log_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c shr_mpi_mod.F90 -o shr_mpi_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c shr_const_mod.F90 -o shr_const_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c shr_sys_mod.F90 -o shr_sys_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c precision_mod.F90 -o precision_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c param_mod.F90 -o param_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c msg_mod.F90 -o msg_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c LICOM_Error_mod.F90 -o LICOM_Error_mod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c POP_BlocksMod.F90 -o POP_BlocksMod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c POP_CommMod.F90 -o POP_CommMod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c POP_SpaceCurveMod.F90 -o POP_SpaceCurveMod.o
mpif90 -m64 -mcmodel=medium -O2 -g -r8 -i4 -free -fp-model precise -convert big_endian -assume byterecl -no-vec -ftz -traceback -O2 -g -r8 -i4 -free -fp-model precise -cc
nvert big_endian -assume byterecl -no-vec -ftz -traceback -I. -I/public/software/mathlib/libs-intel/netcdf/4.4.1/include -I/opt/hpc/software/mpi/hpcx/v2.7.4/intel-2017.5.239
/include -c POP_DistributionMod.F90 -o POP_DistributionMod.o
```





# 新建case、编译、运行 (三)

- DCU版本海洋环流模式为例
  - 第四步：准备输入数据，拷贝/链接到 run 目录下

```
#!/usr/bin/bash
DATAPATH=/public/home/hydeng/LICOM/data/10km

ln -sf ${DATAPATH}/mecator-salt-20160501-3600X2302X55_n2s.nc salt_initial
ln -sf ${DATAPATH}/mecator-temp-20160501-3600X2302X55_n2s.nc temp_initial
ln -sf ${DATAPATH}/chl_clim_monthly_tripole_3600X2302X55_omip_db_20170529.nc MODEL_CHLFRC

ln -sf ${DATAPATH}/ice.2016_daily.nc ice.2016_daily.nc
ln -sf ${DATAPATH}/q_10.2016_daily.nc q_10.2016_daily.nc
ln -sf ${DATAPATH}/rain.2016_daily.nc rain.2016_daily.nc
ln -sf ${DATAPATH}/rlds.2016_daily.nc rlds.2016_daily.nc
ln -sf ${DATAPATH}/rsds.2016_daily.nc rsds.2016_daily.nc
ln -sf ${DATAPATH}/runoff_all.2016_daily.nc runoff_all.2016_daily.nc
ln -sf ${DATAPATH}/slp.2016_daily.nc slp.2016_daily.nc
ln -sf ${DATAPATH}/snow.2016_daily.nc snow.2016_daily.nc
ln -sf ${DATAPATH}/sss_month_phc3_tripole_3600X2302X55_omip_db_20170529.nc sss_phc3_monthly.nc
ln -sf ${DATAPATH}/t_10.2016_daily.nc t_10.2016_daily.nc
ln -sf ${DATAPATH}/tidal_energy_licom_tripole_3600X2302X55_omip_db_20170529.nc tidal_energy.nc
ln -sf ${DATAPATH}/u_10.2016_daily.nc u_10.2016_daily.nc
ln -sf ${DATAPATH}/v_10.2016_daily.nc v_10.2016_daily.nc

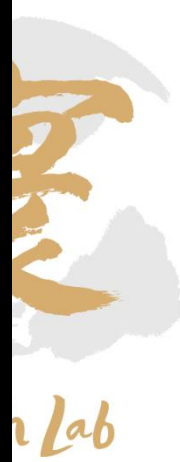
ln -sf ${DATAPATH}/depth.da depth.da
ln -sf ${DATAPATH}/grid.da grid.da
ln -sf ${DATAPATH}/kmt.da kmt.da
ln -sf ${DATAPATH}/dncoef.h1 dncoef.h1
ln -sf ${DATAPATH}/fort.22.0002-01-05 fort.22.0002-01-05
cp ${DATAPATH}/rpointer.ocn rpointer.ocn
cp ${DATAPATH}/ocn.parm ocn.parm
```



# 新建case、编译、运行 (四)

- DCU版本海洋环流模式为例
  - **第五步**: 在 run 目录下, 编辑 “ocn.parm” 文件设置模式参数

```
&namctl
  AM_HOR      = 800
  AH_HOR      = 100
  GAMMA       = 30.0D0
  AM_BIHAR    = -4.0D+8
  AH_BIHAR    = -4.0D+8
  IDTB        = 6
  IDTC        = 120
  IDTS        = 120
  AFB1        = 0.20
  AFC1        = 0.43
  AFT1        = 0.43
  AMV         = 1.0E-3
  AHV         = 0.3E-4
  NUMBER      = 120
  NSTART      = 0
  diag_msf    = .false.
  diag_mth    = .false.
  diag_bsf    = .false.
  IO_HIST     = 1
  IO_REST     = 1
  klv         = 55
  kvt         = 55
  rest_freq   = 1
  out_dir     = "./"
  adv_momentum = "centered"
  adv_tracer   = "tspas"
  boundary_restore = 2 ! 0 for no restoring , 1 for
/
&domain_nml
  nprocs_clinic = 2220
  nprocs_tropic = 2220
  clinic_distribution_type = 'cartesian'
  tropic_distribution_type = 'cartesian'
  ew_boundary_type = 'cyclic'
  ns_boundary_type = 'tripole'
  profile_barrier = .false.
/
&grid_nml
  horiz_grid_file = 'grid.da'
  vert_grid_file = 'depth.da'
  topography_file = 'kmt.da'
  basin_grid_file = 'Basin.data'
  horiz_grid_opt = 'tripole'
/
"ocn.parm" 45L, 1065C
```



# 新建case、编译、运行 (五)

- DCU版本海洋环流模式为例

- 第六步：在 run 目录下，编辑作业脚本
  - 左：run.slurm-DCU (作业脚本)
  - 右：single\_process.sh (DCU版本独有，模拟器绑卡脚本)

```
#!/bin/bash
#SBATCH -p high
#SBATCH -n 2220
#SBATCH --ntasks-per-node=2
#SBATCH --gres=dcu:2
#SBATCH -J licom3-hip
#SBATCH --mem=100G
#SBATCH -o log.%j
#SBATCH -e log.%j
#SBATCH --exclusive

module purge
module load compiler/rocm/3.3
module load compiler/intel/2017.5.239
module load mpi/hpcx/2.7.4/intel-2017.5.239
module load mathlib/netcdf/intel/4.4.1

echo "Start time: `date`" #显示开始时间
echo "SLURM_JOB_ID: $SLURM_JOB_ID" #显示作业号
echo "SLURM_NNODES: $SLURM_NNODES" #显示节点数
echo "SLURM_NTASKS: $SLURM_NTASKS" #显示总任务数
echo "SLURM_TASKS_PER_NODE: $SLURM_TASKS_PER_NODE" #显示每个节点的任务数
echo "SLURM_CPUS_PER_TASK: $SLURM_CPUS_PER_TASK" #显示每个任务使用的cpu数量
echo "SLURM_JOB_PARTITION: $SLURM_JOB_PARTITION" #显示队列/分区名称
echo "SLURM_SUBMIT_DIR: $SLURM_SUBMIT_DIR" #显示提交作业目录的路径
echo "SLURM_NODELIST: $SLURM_NODELIST" #显示执行节点列表名称

which mpirun
hostfile=./$SLURM_JOB_ID
scontrol show hostnames $SLURM_JOB_NODELIST > ${hostfile}
num_node=$(cat $hostfile|sort|uniq|wc -l)

num_DCU=$((num_node*2))
nodename=$(cat $hostfile |sed -n "1p")
dist_url='echo $nodename | awk '{print $1}'`

rm `pwd`/hostfile-dl -f
cat $hostfile|sort|uniq > `pwd`/tmp

for i in `cat ./tmp`
do
    echo ${i} slots=2 >> `pwd`/hostfile-dl
done

#multi-node multi-gpu
mpirun -np $num_DCU --allow-run-as-root --hostfile `pwd`/hostfile-dl --bind-to none
`pwd`/single_process.sh $dist_url
```

```
#!/bin/bash
#
export GLOO_SOCKET_IFNAME=ib0

module load compiler/rocm/3.3
module load compiler/intel/2017.5.239
module load mpi/hpcx/2.7.4/intel-2017.5.239
module load mathlib/netcdf/intel/4.4.1

lrank=$OMPI_COMM_WORLD_LOCAL_RANK
comm_rank=$OMPI_COMM_WORLD_RANK
comm_size=$OMPI_COMM_WORLD_SIZE

#modify by yourself
APP="./licom3.exe"

case ${lrank} in
[0])
    export HIP_VISIBLE_DEVICES=0
    numactl --cpunodebind=4 --membind=0 ${APP}
    ;;
[1])
    export HIP_VISIBLE_DEVICES=1
    numactl --cpunodebind=5 --membind=1 ${APP}
    ;;
esac
~
```





# 新建case、编译、运行 (六)

- DCU版本海洋环流模式为例

- 第七步：运行作业脚本，输入命令“sbatch run.slurm-DCU”

- 模式运行后，一些输出展示见各图

```
stepon begin, version: 10km (GPU/DCU)
test-mm,imd,mon0,month      31      1      1
irec= 5
jra_daily time : 15.1395750045776
(GPU/HIP) stepon test : 24.8522219657898 loops : 480
Mass Imbalance : -2147483648 -2.1474836480000000E-013
in instant,iday=,imd=,mon0=, nwmf=, rest_freq= 5 31
1 2 1
in instant,write_restart T
fname=fort.22.0002-01-06
day: 57.73282909393105 SYPD: 4.100133883651968 1 5
irec= 6
jra_daily time : 6.31282997131348
(GPU/HIP) stepon test : 24.3122110366821 loops : 480
Mass Imbalance : -2147483648 -2.1474836480000000E-013
in instant,iday=,imd=,mon0=, nwmf=, rest_freq= 6 31
1 2 1
in instant,write_restart T
fname=fort.22.0002-01-07
day: 41.658293008804321 SYPD: 5.682237837183944 1 6
irec= 7
jra_daily time : 6.33890604972839
(GPU/HIP) stepon test : 22.6857750415802 loops : 480
Mass Imbalance : -2147483648 -2.1474836480000000E-013
in instant,iday=,imd=,mon0=, nwmf=, rest_freq= 7 31
1 2 1
in instant,write_restart T
fname=fort.22.0002-01-08
day: 40.336492061614990 SYPD: 5.868441122880476 1 7
irec= 8
jra_daily time : 6.33290100097656
(GPU/HIP) stepon test : 20.8253300189972 loops : 480
Mass Imbalance : -2147483648 -2.1474836480000000E-013
in instant,iday=,imd=,mon0=, nwmf=, rest_freq= 8 31
1 2 1
in instant,write_restart T
fname=fort.22.0002-01-09
day: 38.534972906112671 SYPD: 6.142792142188698 1 8
irec= 9
jra_daily time : 6.32986497879028
(GPU/HIP) stepon test : 20.6694800853729 loops : 480
Mass Imbalance : -2147483648 -2.1474836480000000E-013
in instant,iday=,imd=,mon0=, nwmf=, rest_freq= 9 31
1 2 1
in instant,write_restart T
fname=fort.22.0002-01-10
day: 38.225443124771118 SYPD: 6.192533281941925 1 9
irec= 10
jra_daily time : 6.3244710405645
```

```
Start time: Fri Sep 3 18:37:38 CST 2021
SLURM_JOB_ID: 156334
SLURM_NNODES: 1110
SLURM_NTASKS: 2220
SLURM_TASKS_PER_NODE: 2(x1110)
SLURM_CPUS_PER_TASK:
SLURM_JOB_PARTITION: high
SLURM_SUBMIT_DIR: /public/home/hydeng/LICOM/run-2220
```

```
jra_daily time : 6.31085181236267
(GPU/HIP) stepon test : 20.7950911521912 loops : 480
Mass Imbalance : -2147483648 -2.1474836480000000E-013
in instant,iday=,imd=,mon0=, nwmf=, rest_freq= 31 31
2 2 1
in instant,write_restart T
fname=fort.22.0002-02-01
day: 38.162539958953857 SYPD: 6.202740410405645 1 31
End time: Fri Sep 3 18:57:24 CST 2021
```

```
lrwxrwxr-x 1 hydeng hydeng 54 Sep 3 11:26 fort.22.0002-01-05 -> /public/home
/hydeng/LICOM/data/10km/fort.22.0002-01-05
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:40 fort.22.0002-01-06
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:40 fort.22.0002-01-07
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:41 fort.22.0002-01-08
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:42 fort.22.0002-01-09
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:42 fort.22.0002-01-10
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:43 fort.22.0002-01-11
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:44 fort.22.0002-01-12
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:44 fort.22.0002-01-13
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:45 fort.22.0002-01-14
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:46 fort.22.0002-01-15
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:46 fort.22.0002-01-16
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:47 fort.22.0002-01-17
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:48 fort.22.0002-01-18
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:48 fort.22.0002-01-19
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:49 fort.22.0002-01-20
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:49 fort.22.0002-01-21
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:50 fort.22.0002-01-22
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:51 fort.22.0002-01-23
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:52 fort.22.0002-01-24
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:52 fort.22.0002-01-25
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:53 fort.22.0002-01-26
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:54 fort.22.0002-01-27
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:54 fort.22.0002-01-28
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:55 fort.22.0002-01-29
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:56 fort.22.0002-01-30
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:56 fort.22.0002-01-31
-rw-rw-r-- 1 hydeng hydeng 3712666048 Sep 3 18:57 fort.22.0002-02-01
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:40 fort.23.0002-01-06
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:40 fort.23.0002-01-07
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:41 fort.23.0002-01-08
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:42 fort.23.0002-01-09
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:42 fort.23.0002-01-10
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:43 fort.23.0002-01-11
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:44 fort.23.0002-01-12
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:44 fort.23.0002-01-13
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:45 fort.23.0002-01-14
-rw-rw-r-- 1 hydeng hydeng 3646368440 Sep 3 18:46 fort.23.0002-01-15
```



# 新建case、编译、运行 (七)

- CPU版本海洋环流模式
  - 第一步: load相应必要的软件包、库、编译环境等
    - module load compiler/intel/2017.5.239
    - module load mpi/hpcx/2.7.4/intel-2017.5.239
    - module load mathlib/netcdf/intel/4.4.1
    - ~~module load compiler/rocm/3.3~~
  - 第二步: 在 src 目录下, 编辑 “def-undef.h” 文件, 确定选用的功能, 设置网格划分方案
  - 第三步: 在 src 目录下, 输入 “make” 命令, 编译程序
  - 第四步: 准备输入数据, 拷贝/链接到 run 目录下
  - 第五步: 在 run 目录下, 编辑 “ocn.parm” 文件设置模式参数
  - 第六步: 在 run 目录下, 编辑作业脚本
    - run.slurm (作业脚本)
    - ~~single\_process.sh (DCU版本独有, 模拟器绑卡脚本)~~
  - 第七步: 运行作业脚本, 输入命令 “sbatch run.slurm”



# 系统常见问题处理

现象	可能原因	解决方案
算例创建过程中，报缺少inputdata相关数据	<ol style="list-style-type: none"><li>1. 系统配置时，inputdata路径设置不正确</li><li>2. 数据文件在设定的系统inputdata文件夹下没有</li></ol>	<ol style="list-style-type: none"><li>1. 重新设置inputdata路径</li><li>2. 将输入数据准备并拷贝到inputdata设置的路径</li></ol>
算例创建过程中，报可执行程序没有权限	没设置可执行程序/输入数据权限	可通过chmod命令修改可执行程序文件权限为“可执行”，形式为“chmod +x 可执行文件名”
算例运行时，报“segment fault”错误	可能与机器操作系统配置有关，常遇到的问题是堆栈（stack）设置过小	使用limit命令可以修改相应配置
算例运行时，报运行进程数和程序设置所需进程数不匹配	应该是模式并行资源分配和运行申请进程数不匹配	查看并修改“def-undef.h”“ocn.parm”和作业脚本文件，确认设定的进程数一致
算例运行时，报读取文件错误	<ol style="list-style-type: none"><li>1. 输入文件有问题，如数据格式、大小、正确性等</li><li>2. 预编译功能开关设置出现冲突</li></ol>	<ol style="list-style-type: none"><li>1. 检测输入文件正确性</li><li>2. 检测“def-undef.h”文件，确保功能选项没发生冲突</li></ol>



谢 谢