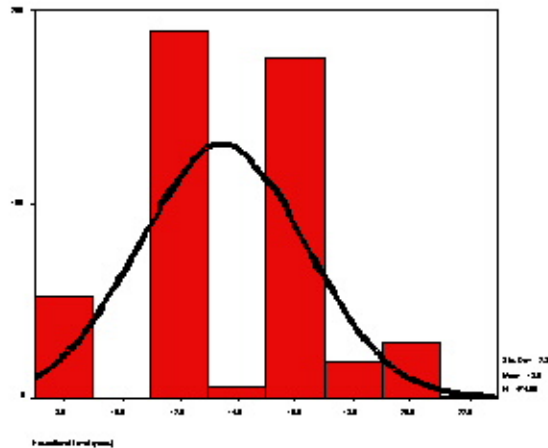


# ITC Research Computing Support Center



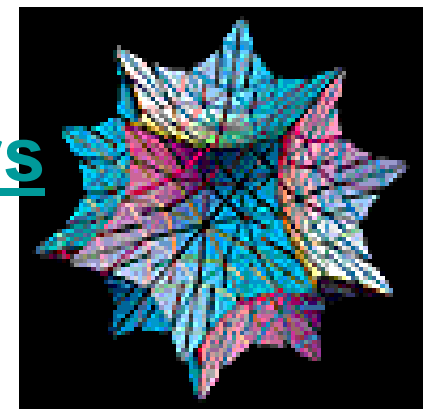
**Welcome**  
**Help yourself to the snacks and drinks. Please sign in and get a handout & feedback form**

Phone: 243-8800

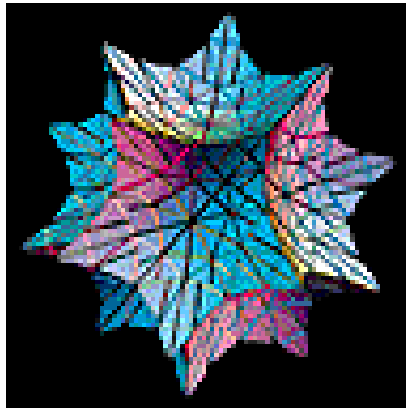
E-Mail: Res-Consult@Virginia.EDU

<http://www.itc.Virginia.edu/researchers>

Fax: 243-8765



# ITC Research Computing Support Implementing and Debugging Parallel Codes



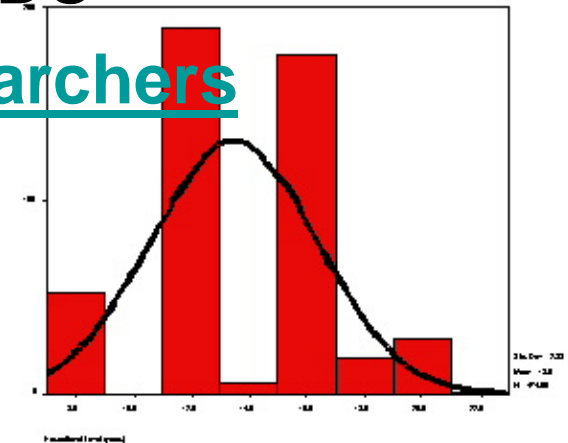
By: Katherine Holcomb

Research Computing Support Center

Phone: 243-8800 ☎ Fax: 243-8765

E-Mail: [Res-Consult@Virginia.EDU](mailto:Res-Consult@Virginia.EDU)

<http://www.itc.Virginia.edu/researchers>



# Outline

- Some advanced features of MPI
  - Nonblocking send/recv
  - MPI types
  - Cartesian topologies
  - Some help with profiling
- Example: Barotropic Ocean Model
- Debugging with the Totalview debugger

# Nonblocking Send/Recv

```
real, dimension(:,,:), allocatable :: a, b
integer req(4)
integer status(MPI_STATUS_SIZE,4)
....
call MPI_COMM_SIZE(comm, p, ierr)
call MPI_COMM_RANK(comm,myrank,ierr)
m=n/p
if (myrank .lt. (n-p*m)) then
    m=m+1
endif
```

```
if (myrank.eq.0) then
  left = MPI_PROC_NULL
else
  left = myrank -1
endif
```

```
if (myrank .eq. p-1) then
  right = MPI_PROC_NULL
else
  right = myrank+1
endif
```

```
allocate (a(0:,n+1, 0:m+1), b(n,m))
```

```
do while (.not. converged)
  do i=1, n
    b(i,1) = 0.25*(a(i-1,1) + a(i+1,1) + a(i,0) + a(i,2))
    b(i,m) = 0.25*(a(i-1,m) + a(i+1,m) + a(i,m-1) + a(i,m+1))
  end do
```

```
call MPI_Irecv(a(1,0),n, MPI_REAL, left, tag, comm, req(3), ierr)
call MPI_Irecv(a(1,m+1),n,MPI_REAL,right,tag,comm,req(4), ierr)
```

```
call MPI_Isend(b(1,1), n, MPI_REAL, left, tag, comm, req(1), ierr)
call MPI_Isend(b(1,m), n, MPI_REAL, right, tag, comm, req(2), ierr)
```

```
do j=2, m-1
  do i=1,n
    b(i,j) = 0.25*(a(i-1,j)+a(i+1,j) + a(i,j-1)+a(i,j+1))
  enddo
enddo
```

```
do j=1,m
  do i=1,n
    a(i,j) = b(i,j)
  enddo
enddo
```

```
do i=1,4
  call MPI_WAIT(req(i), status(1,i), ierr)
enddo
```

# MPI Types

- Derived datatype
- Makes it possible to send non-contiguous data in a single message
  - rows in Fortran, columns in C (et al.)
- Also provides a mechanism to send non-homogeneous data in a single message
  - structures or types

# Several Predefined Types Exist

- MPI\_Type\_indexed
  - for matrix sections
- MPI\_Type\_vector
  - can be used for rows or columns
- MPI\_Type\_struct
  - for inhomogeneous data

# Example: Vector Type

- Constructor:
  - `MPI_Type_Vector(count, blocklength, stride, oldtype, newtype [,ierr])`
- Must be committed before it can be used
  - `MPI_Type_Commit(datatype [,ierr])`
- General rule: Fortran uses subroutines and `ierr` is a parameter; C returns `ierr` as the function value, otherwise the parameter lists are the same (but watch for ampersands in the C/C++ parameter lists)

# Creating a Vector Type



call `mpi_type_vector(myny,1,mynx,MPI_REAL,strided, ierr)`

call `mpi_type_commit(strided,ierr)`

# Cartesian Topologies

- If the layout of the communicating processes is simple, e.g. two- or three-dimensional grids of, MPI can create the mapping process->hardware for you.

0 (0,0)	1 (0,1)	2 (0,2)	3 (0,3)	4 (0,4)
5 (1,0)	6 (1,1)	7 (1,2)	8 (1,3)	9 (1,4)
10 (2,0)	11 (2,1)	12 (2,2)	13 (2,3)	14 (2,4)

`MPI_CART_CREATE(comm_old, ndims, dims, periods, reorder, comm_cart)`

INPUT: `comm_old` input communicator (often `MPI_COMM_WORLD`)  
`ndims` number of dimensions of Cartesian process grid  
`dims` integer array of size `ndims` with number of processes  
along each dimension  
`periods` logical array of size `ndims` specifying whether the grid  
is periodic (edges wrap) or not (edges have no neighbor)  
`reorder` ranks may be reordered or not

OUTPUT `comm_cart` communicator with new Cartesian topology

```
ndims = 2  
dims(1) = 5  
dims(2) = 3  
periods(1) = .true.  
periods(2) = .false.  
reorder = .false.
```

```
call MPI_CART_CREATE(MPI_COMM_WORLD, ndims, dims, periods, reorder, &  
                    comm2d)
```

# More Cartesian Functions

- `MPI_Cart_get(comm, maxdims, dims, periods, coords [,ierr])`
  - Returns information on the topology
- `MPI_Cart_rank(comm, coords, rank [,ierr])`
  - Translates from cartesian numbering to rank order
- `MPI_Cart_coords(comm, rank, maxdims, coords [,ierr])`
  - Translates from rank order to cartesian numbering
- `MPI_Cart_shift(comm, direction, disp, rank_source, rank_dest [,ierr])`
  - Returns information needed to call `sendrecv` in the given direction

# Profiling

- Profiling an MPI code usually requires hand accumulation of the time.
- `MPI_WTIME()`
  - returns a floating-point (double precision) number of seconds, representing elapsed wall-clock time from some previous starting point.

```
start = MPI_Wtime();  
---- do computations/communications  
end= MPI_Wtime();  
etime = end-start;
```

# EXAMPLE

## Barotropic Ocean Model

- A barotropic model solves the “shallow-water” equations

$$\partial_t U = fV - gH(\partial_x \eta) + (\tau_w - \tau_b)_x + A\nabla^2 U - \partial_x(UV/H)$$

$$\partial_t V = -fU - gH(\partial_y \eta) + (\tau_w - \tau_b)_y + A\nabla^2 V - \partial_y(UV/H)$$

$$\partial_t \eta = -(\partial_x U + \partial_y V)$$

The barotropic equations give the depth-averaged mode of the circulation. This is of interest for two main reasons:

1. The free-surface elevation (sea-surface height or SSH) is mostly determined by this mode. This includes tides (with appropriate forcings and topographies taken into account).
2. “Real” ocean models solve the barotropic mode separately from the depth-dependent (baroclinic) modes.

# Numerical Solution

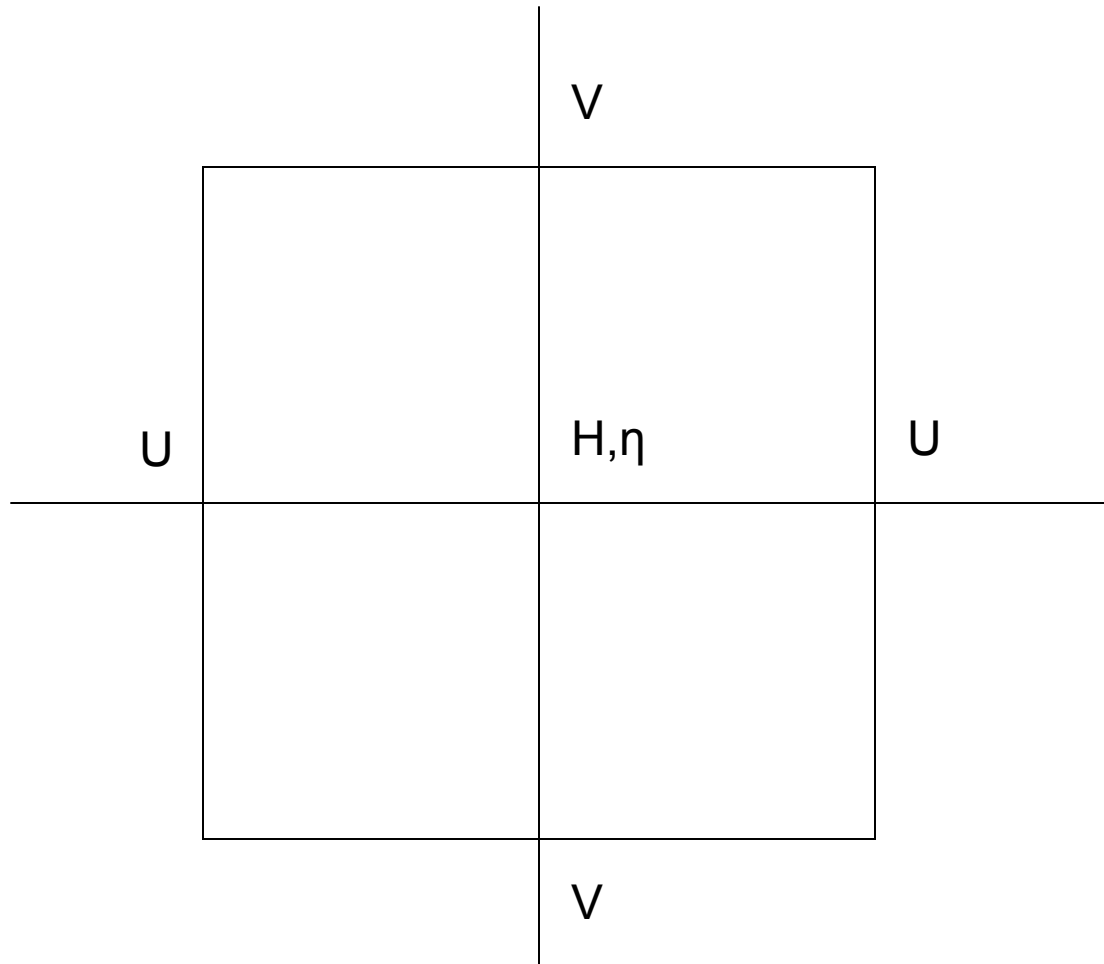
- Our simple model will use the leapfrog method (centered differences)

$$(U^{n+1}_{i,j} - U^{n-1}_{i,j})/\Delta t = fV^n_{i,j} - ((gH)/\Delta x)(\eta^n_{i,j} - \eta^n_{i-1,j}) + (A/\Delta x^2)(U^{n-1}_{i+1,j} + U^{n-1}_{i-1,j} + U^{n-1}_{i,j+1} + U^{n-1}_{i,j-1} - 4U^{n-1}_{i,j}) + (\tau_x)^n_{i,j} - C_d U^{n-1}_{i,j}$$

$$(V^{n+1}_{i,j} - V^{n-1}_{i,j})/\Delta t = -fU^n_{i,j} - ((gH)/\Delta y)(\eta^n_{i,j} - \eta^n_{i-1,j}) + (A/\Delta y^2)(V^{n-1}_{i+1,j} + V^{n-1}_{i-1,j} + V^{n-1}_{i,j+1} + V^{n-1}_{i,j-1} - 4V^{n-1}_{i,j}) + (\tau_y)^n_{i,j} - C_d V^{n-1}_{i,j}$$

$$(\eta^{n+1}_{i,j} - \eta^{n-1}_{i,j})/\Delta t = (U^n_{i+1,j} - U^n_{i,j})/\Delta x - (V^n_{i,j+1} - V^n_{i,j})/\Delta y$$

# Grid



0	1
2	3

# Communications

- Create an `MPI_TYPE_VECTOR` to send the rows (for Fortran)
- Use `MPI_CART_CREATE` to set up the neighbors for a given number of processes (4 in this example)
- Send boundary data above/below, left/right (not periodic)

# Running the Example

- In `/export/rescomp/mpi_example` on Cedar ([cedar.itc.virginia.edu](http://cedar.itc.virginia.edu))
- Read the README before beginning

# Debugging with Totalview

- Use the frontend only for short debugging runs
- mpiexec can be used only under PBS – you must use mpirun on the frontend
- Load the module corresponding to your compiler
  - e.g. module load mpich-eth-intel
- Create a file called something like “host” consisting of the line  
localhost :4

# Running Totalview

- You must have an X server installed on your local machine
    - The X server runs on the machine doing the display (your desktop). The client runs on the remote system.
  - eXceed : commercial software, cost is \$35 for Uva-owned machine through RCSC
  - Cygwin/Xorg : free (and works fine)
    - [www.cygnum.com](http://www.cygnum.com) (be sure to add Xorg in setup)
    - to use without installing to disk, get the XLiveCD from the University of Indiana  
xlivecd.indiana.edu (get ISO image and burn to CD)
- Use via X11 forwarding
- eXceed : use SecureCRT, enable X11 port forwarding
  - Xorg: use bash shell to log on to frontend with `ssh -X yourid@cedar.itc.virginia.edu`

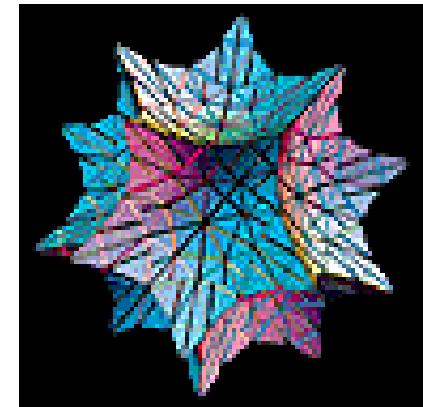
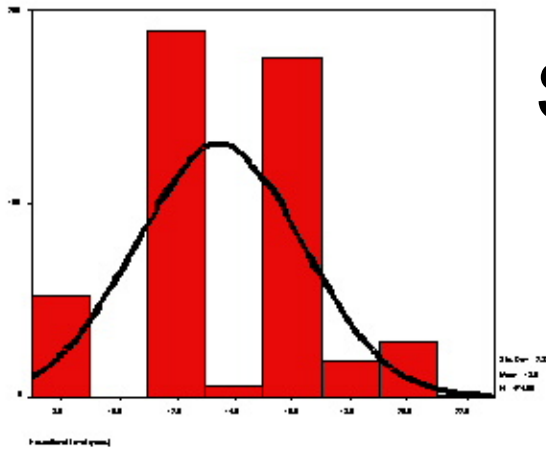
# Starting Totalview

- To use Totalview with an MPI program, it must be started by the MPI executor (module loaded, X server running locally):

```
mpirun -np 4 -dbg=totalview mybinary
```

- If the program crashes, or otherwise halts without calling `MPI_Abort` or `MPI_Finalize`, then  
killall mybinary  
cleanipcs

## Some Useful Information



Cedar Hands-on Tutorial is online at [www.itc.virginia.edu/research/linux-clusters/cedar/hands-on](http://www.itc.virginia.edu/research/linux-clusters/cedar/hands-on)

Totalview Information at [www.itc.virginia.edu/research/totalview](http://www.itc.virginia.edu/research/totalview)

• Talks are online at [www.itc.virginia.edu/research/talks](http://www.itc.virginia.edu/research/talks)