Parallelization of the FVCOM Code: Application to the GOM/GB

G Cowles, C. Chen, H. Liu, B. Rothschild

US GLOBEC Georges Bank Science Meeting

18th - 20th November 2003

Motivation

Computational Cost of Serial FVCOM Model

- High Resolution Studies Require Large Compute Times
- Forecasting Effectiveness is Limited if
  Compute Time $>$ Forecast Period

Parallel Machines becoming Accessible

- Low Cost of Commodity Based Clusters (Beowulfs)
- Dual Processor SMP's Common as Desktops

Procedure

Implementation Design

- Prescribe a division of Work among Processing Elements (Paradigm Choice)
- Utilize Parallel Languages to Power this Division (Model Choice)

Must Take into Account:

- Baseline Serial Algorithm (FVCOM)
- Time/Effort Available
- Target Parallel Machine Architecture(s)

Methodology Implemented

Programming Abstraction: Single Program/Multiple Data (SPMD)

- Each Proc Integrates Equations in SubDomain using FVCOM
- Along SubDomain Boundaries, Data is Exchanged w/MPI
- Data is collected to Global Arrays for output

Work Decomposition: Physical Subdomains

- METIS Graph-Partitioner
- Horizontal Decomposition Only

Programming Model: Message Passing (MPI 2.0)

- MPI 2.0 Libraries
- Highly Portable

Main Parallel Routines

- There are four major constructs needed for parallelizing within the SPMD framework.
- In the FVCOM application, the *Exchange* routines must be streamlined and efficient.

Construct	Description	Transform	MPI Comm Type
Decompose	Subdivide Global Dom	G => L	–
Exchange	Proc Boundary Update	L => L	Non Blocking Sends/Recvs
Collect	Global Ouput/Report	L => G	Blocking Send/Recv
Disperse	Local IC's/BC's	G => L	Blocking Send/Recv

FVCOM Code Skeletons

Read Parameters
Read Grid/Connectivity

Read Initial Conditions

Main Loop
External Mode Loop
Runge-Kutta Loop

Update Eta
Update UA/VA

Adjust 3-D Velocity Field
Update T/S
Update Q/Q2
Update U/V

Output Results
Close Up

Serial

Read Parameters
Read Grid/Connectivity
Decompose Domain
Read Initial Conditions
Disperse IC/BC to Local Domains

Main Loop
External Mode Loop
Runge-Kutta Loop

Update Eta
Update UA/VA
Exchange (Eta,UA,VA)

Adjust 3-D Velocity Field
Update T/S
Update Q/Q2
Update U/V
(Exchange U,V,S,T,Q,Q2)

Collect Flowfield to Global

Output Results
Close Up

Parallel

Domain Decomposition

Goals:

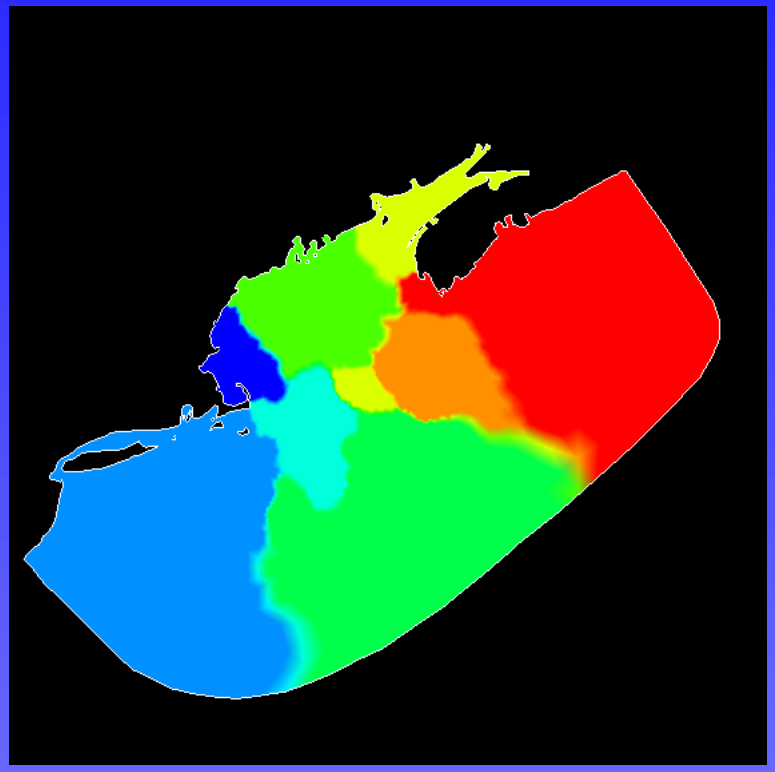
1.) Generate Partitions of equal size - balance the load

2.) Minimize interprocessor boundary length - minimize comm

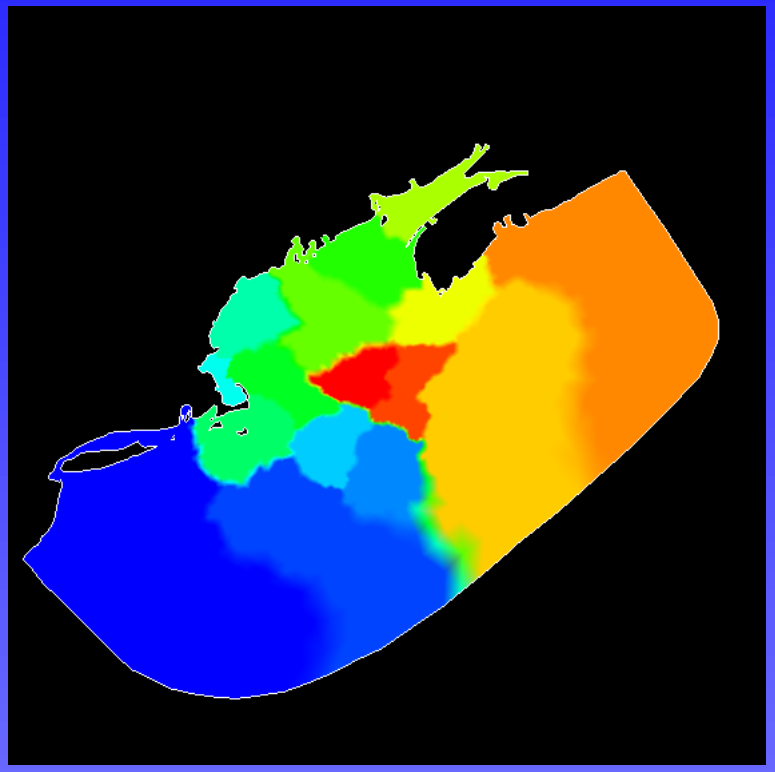
Solution: METIS - Graph partitioning libraries

- Input: grid connectivity + number of desired partitions
- Output: partition map
- decomposition time is trivial w.r.t. model computation time
- load imbalance with 16 procs: 3%

Domain Decomposition: Partitions of Georges Bank/GOM Using METIS

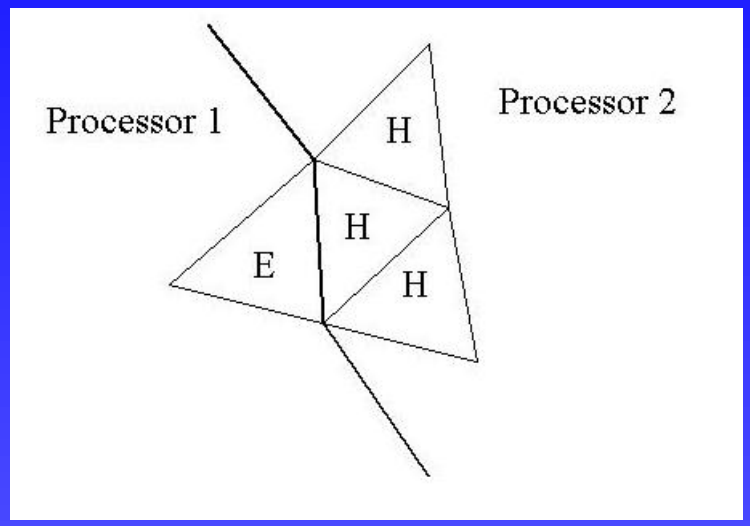


8 Way



16 Way

Data Exchange

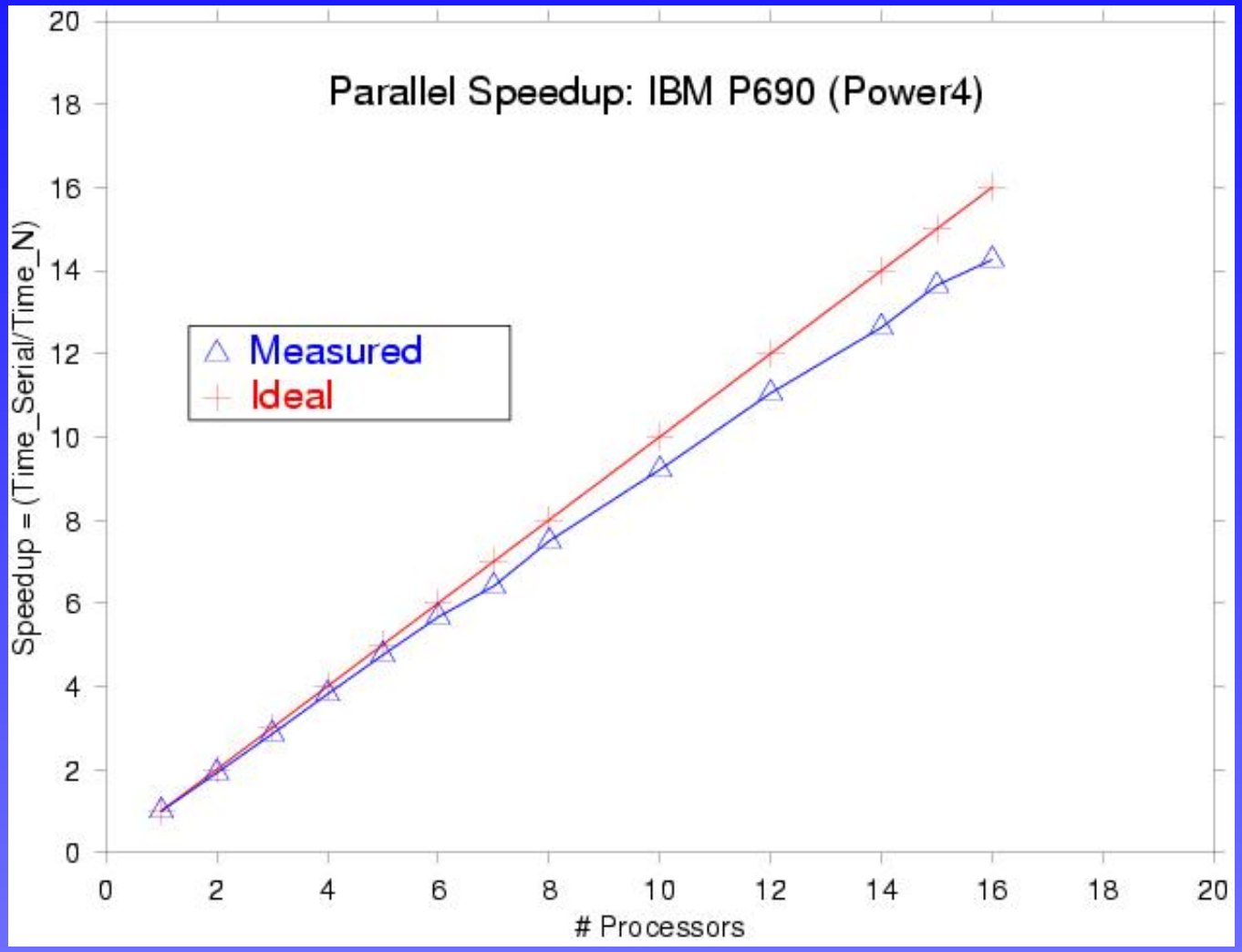


Flux Stencil for
Element-Based Quantity

To Update Element "E" In Proc 1:

- Values of Quantities Halo Elements Must be Updated
- Exchange of Data is Performed Between Procs 1 and 2
- A Compact Stencil minimized Amount of Data to be Xchanged
- Node Based Values have a different Stencil

MultiProcessor Performance



Other Modifications

- Migration to F95 with F2K Extensions - *Readability*
- Allocatable Memory - *Robustness, Usability*
- Format-Free Input File - *Robustness*
- Float Choice using Selected_Real_Kind - *Portability*
- Streamlining of Core Subroutines - *Speed*
- NetCDF Input/Output - *Portability, Robustness, Usability*
- Preprocessing Directives - *Promotes Modular Coding/Portability*

Runtimes

Machine	Code	Procs	Sec/It	1 Month Comp Time
Dell P4/2.40 GHz	Orig	1	10.57	2d 15h 30m
Dell P4/2.40 GHz	Parallel	1	6.46	1d 14h 40m
Dell P4/2.40 GHz	Parallel	2	4.05	1d 00h 00m
IBM Regatta/PWR4	Parallel	1	4.42	1d 02h 33m
IBM Regatta/PWR4	Parallel	4	1.16	06h 56m
IBM Regatta/PWR4	Parallel	8	.59	03h 31m
IBM Regatta/PWR4	Parallel	16	.31	01h 51m

Runtimes for Gulf of Maine/George's Bank Model
(25K Ω, 31 σ Levels, ΔT = 120 Seconds)

Application: GOM/GB Simulation

Computation	Elements	σ Layers	ΔT	Integration Period
Prognostic Sim	25K	31	120 sec	1 Year
Data Assimilated	25K	31	120 sec	1 Year x 2

Model Summary

Task	cpu hours	Procs	Wall Clock
DA Model Run	1330	12 Power4	108h
Model Run	530	4 Pentium4	132h
Data Transfer	40	1 Pentium4	40h
LP Filter	120	1 Pentium4	120h

Work Requirements

Conclusions

- Parallelization of FVCOM core code completed
- Speedup indicates implementation is efficient
- Monthly Integrations of our GOM/GB model reduced from 63 to 2 hours.
- Year Integration of GOM/GB completed in 2 Weeks
- Enables application of FVCOM to simulations requiring higher resolution/larger time scales
- Facilitates integration of FVCOM into regional forecasting systems

Acknowledgements

NASA

- Funding through Regional Fisheries Application Center (RFAC)

Prof Chen's Group (MEDM)

- FVCOM Help (Hedong Liu)

ARSC

- Computational Facilities
- Coding Advice (Kate Hedström)