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.

<table>
<thead>
<tr>
<th>Construct</th>
<th>Description</th>
<th>Transform</th>
<th>MPI Comm Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decompose</td>
<td>Subdivide Global Dom</td>
<td>G =&gt; L</td>
<td>–</td>
</tr>
<tr>
<td>Exchange</td>
<td>Proc Boundary Update</td>
<td>L =&gt; L</td>
<td>Non Blocking Sends/Recvs</td>
</tr>
<tr>
<td>Collect</td>
<td>Global Output/Report</td>
<td>L =&gt; G</td>
<td>Blocking Send/Recv</td>
</tr>
<tr>
<td>Disperse</td>
<td>Local IC’s/BC’s</td>
<td>G =&gt; L</td>
<td>Blocking Send/Recv</td>
</tr>
</tbody>
</table>
FVCOM Code Skeletons

Serial

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

Parallel

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
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

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

Parallel Speedup: IBM P690 (Power4)

Speedup = (Time_SERIAL / Time_N)

- Measured
- Ideal

# Processors
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

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

Runtimes for Gulf of Maine/George’s Bank Model

(25K Ω, 31 σ Levels, ΔT = 120 Seconds)
Application: GOM/GB Simulation

<table>
<thead>
<tr>
<th>Computation</th>
<th>Elements</th>
<th>$\sigma$ Layers</th>
<th>$\Delta T$</th>
<th>Integration Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prognostic Sim</td>
<td>25K</td>
<td>31</td>
<td>120 sec</td>
<td>1 Year</td>
</tr>
<tr>
<td>Data Assimilated</td>
<td>25K</td>
<td>31</td>
<td>120 sec</td>
<td>1 Year x 2</td>
</tr>
</tbody>
</table>

Model Summary

<table>
<thead>
<tr>
<th>Task</th>
<th>cpu hours</th>
<th>Procs</th>
<th>Wall Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA Model Run</td>
<td>1330</td>
<td>12 Power4</td>
<td>108h</td>
</tr>
<tr>
<td>Model Run</td>
<td>530</td>
<td>4 Pentium4</td>
<td>132h</td>
</tr>
<tr>
<td>Data Transfer</td>
<td>40</td>
<td>1 Pentium4</td>
<td>40h</td>
</tr>
<tr>
<td>LP Filter</td>
<td>120</td>
<td>1 Pentium4</td>
<td>120h</td>
</tr>
</tbody>
</table>

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)