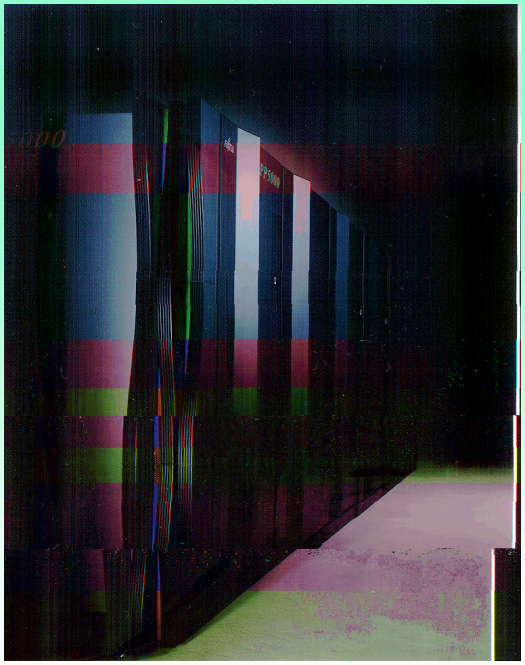
Pilfered Presentation on Progress in Polar Programs

Philip Jones
Los Alamos National Laboratory
Boulder, Colorado
February, 2003

Los Alamos
NATIONAL LABORATORY

Vector Computing
&
Status of Tuning CICE 3.1



Clifford Chen
Fujitsu America, Inc.
San Jose, California
January, 2003

Los Alamos
NATIONAL LABORATORY

Characteristics of CICE3.1 - From Vectorization Point of View

- The Good
  - Theory is well documented
  - Code is neatly written with plenty of comments and easy to understand
- The Bad
  - Short vector length – limited by the grid size
  - CPU distribution is relatively flat
  - The most inner do-loop length is the shortest
- The Ugly
  - Heavy in IF branch check due to the nature of physics

Top 15 Most Expensive Program Units

Procedure List:

Count	Percent	VL	V_Hit(%)	Name
1030	100.0(22.6)	-	50.5	ice_atmo.stability_
626	100.0(13.7)	76	11.7	ice_therm_cice.thermo_cice_
588	67.6(12.9)	112	99.5	ice_dyn_evp.stepu_
505	49.4(11.1)	2006	0.2	ice_transport_remap.flux_integrals_
278	32.0(6.1)	100	100.0	ice_dyn_evp.stress_
342	57.3(7.5)	40	90.4	ice_itd.check_state_
167	89.3(3.7)	100	6.0	ice_mechred_cice.ridge_shift_
117	11.4(2.6)	15	17.1	ice_transport_remap.triangle_coordinates_
116	11.4(2.5)	719	4.3	ice_transport_remap.load_tracers_
113	18.9(2.5)	36	2.7	ice_itd.aggregate_
108	10.6(2.4)	4	0.9	ice_transport_remap.departure_points_
95	100.0(2.1)	53	63.2	ice_albedo.albedos_
64	6.3(1.4)	100	98.4	ice_transport_remap.limited_gradient_
44	4.3(1.0)	100	25.0	ice_transport_remap.update_fields_
40	6.7(0.9)	51	7.5	ice_itd.shift_ice_

Los Alamos
NATIONAL LABORATORY

Original Program stability

- Compute coefficients for atm/ice fluxes, stress, and reference temperature.
- Original code took 10.28 seconds which was 22.6% of total CPU time. Code was running in scalar speed.
- True Ratio of IF statement was less than 2%.
- No. of arrays (vectors) involved: more than 15

Original code

```
!initialization
      . . . .
      do j = jlo,jhi
        do i = ilo,ihi
          if ( (sfctype(1:3)=='ice' .and. aicen(i,j,n) > puny)
     &                           .or.
     &         (sfctype(1:3)=='ocn' .and. tmask(i,j)) ) then
            . . . .
!iterate to converge on Z/L, ustar, tstar and qstar
            do k=1,5
             . . . .
            enddo
            . . . .
          endif
        enddo
      enddo
```

1

2

3

4

Los Alamos NATIONAL LABORATORY

Tuned stability

- IF check was isolated.
- More data movement in the tuned code
- Vector length (itrue) was increased from 50 to more than 1024 for ice case and 8006 for ocean case.
- CPU time reduced from 10.28 to 0.2 seconds on VPP5000.
- Memory accessing pattern was not altered from hardware point of view.
- Bill cleaned up the code and

Los Alamos
NATIONAL LABORATORY

Original Program stepu

- Calculates surface stresses and integrates momentum equation to find velocity (u,v).
- Original code took 6.24 seconds which was 13.3% of total CPU time. Code was fully vectorized with 99.7 vector hit rate.
- True Ratio of IF statement was less than 2%.
- Number of arrays involved: less than 8

Original code

```
do j=jlo,jhi
 do i=ilo,ihi
  if (icetmask(i,j)) then
    . . . .   !stresses for momentum equation
  else
    str = 0.0
  endif
 enddo
enddo
call bound_narr_ne(8,str)
do j=jlo,jhi
 do i=ilo,ihi
  if (iceumask(i,j)) then
    . . . .   !integrate momentum equation
    . . . .   !calculate u, v, strintx,y and strocnx,y
  else
    . . . .   !set u,v, strintx,y and strocnx,y to zero
  endif
 enddo
enddo
```

1

2

Los Alamos NATIONAL LABORATORY

First Tuning on stepu

- The first do-loop was vectorized in j-direction, which caused inefficiency in data access. This can be improved with compiler directive to vectorize the do-loop in i-direction.
- Insert compiler directive to vectorize if statement of the second loop with list vector method.
- CPU time reduced from 6.24 to 2.31 seconds.

```
      do j=jlo,jhi
!ocl novrec
       do i=ilo,ihi
        if (icetmask(i,j)) then
          . . . .   !stresses for momentum equation
        else
          str = 0.0
        endif
       enddo
      enddo
      call bound_narr_ne(8,str)
      do j=jlo,jhi
!ocl vec(list)
       do i=ilo,ihi
        if (iceumask(i,j)) then
          . . . .   !integrate momentum equation
          . . . .   !calculate u, v, strintx,y and strocnx,y
        else
          . . . .   !set u,v, strintx,y and strocnx,y to zero
        endif
       enddo
      enddo
```

1

2

Los Alamos NATIONAL LABORATORY

Second Tuning on stepu

- Isolate IF check of icetmask and iceumask. The vector length increased from 100 to more than 1024.
- The u and v of current step depends on u and v of previous step, hence working temporary arrays for u and v are needed.
- CPU time reduced further down to 1.89 seconds.
- The results were not blessed yet.

Future Tuning Work

- Continue pushing for better vectorization on other program units.
- Reduce redundant IF check by using global variables to store repeated IF branch count.
- Carry out strength reduction on scalar operation.
  - Pay attention to numerical effect.

```
TsfK = Tsf(i,j) + Tffresh ! surface temp (K)
qsat    = qqq / exp(TTT/TsfK) ! saturation humidity (kg/m^3)
dqsatdt = (TTT / TsfK**2) * qqq / exp(TTT/TsfK) ! d(qsat)/dT
```

```
TsfK = 1.0d0/(Tsf(i,j) + Tffresh) ! surface temp (K)
qsat    = qqq / exp(TTT*TsfK) ! saturation humidity (kg/m^3)
dqsatdt = (TTT * TsfK*TsfK) * qqq / exp(TTT*TsfK) ! d(qsat)/dT
```

Reduces three divisions to one inverse and three multiplications

Los Alamos
NATIONAL LABORATORY

Summary

- Current CICE3.1 has been gone through rigorous diagnostic process.
- We obtained more than 50 times speedup on stability after it was well vectorized.
- The tuned stability gained 10% improvement on scalar machine.
- Even a well vectorized subroutine like stepu, we still could push more than 3 times speedup.
- Vectorization for CICE is feasible.

Los Alamos
NATIONAL LABORATORY