Early Results for a DCA-QMC Code

James B. White III (Trey), CCS

John Levesque, Cray

Acknowledgement

Research sponsored by the Mathematical, Information, and Computational Sciences Division, Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

DCA-QMC

- Distributed Cluster Approximation, Quantum Monte Carlo
- Developed by Mark Jarrell *et al.*
- Code, input sets, and sample output provided by Jim Hague

Optimization strategy

- Profile code on IBM p690 (Cheetah)
- Generate loopmarks on SV1
- Vectorize top routines in IBM profile
  - Minimize code modification
- Run on SV1 MSP (4 SSPs)
- Provide to John to run on X1 MSP

Problem loops (meas.F)

```
45.  1---------------<        do ic=1,Nc
46.  1 2-------------<         do jc=1,Nc
47.  1 2 3-----------<          do n=1,nl
48.  1 2 3 Vp--------<            do m=1,nl
49.  1 2 3 Vp                     k=n-m
50.  1 2 3 Vp                     uno=halfr
51.  1 2 3 Vp                     if(k.lt.0) then
52.  1 2 3 Vp                       uno=-halfr
53.  1 2 3 Vp                       k=k+nl
54.  1 2 3 Vp                     end if
55.  1 2 3 Vp                     k=k+1
56.  1 2 3 Vp                     i=icrdiff(ic,jc)
57.  1 2 3 Vp                     ind=ic + (n-1)*Nc
58.  1 2 3 Vp                     imd=jc + (m-1)*Nc
59.  1 2 3 Vp r-----<>               data(k,i)=data(k,i)+(gu(ind,imd)+gd(ind,imd))*uno
60.  1 2 3 Vp-------->            end do
61.  1 2 3----------->          end do
62.  1 2------------->         end do
63.  1--------------->        end do
```

Vp = Parial vectorization = BAD

Problem loops (meas.F)

ftn-6254 f90: VECTOR File = meas.F, Line = 45
A loop starting at line 45 was not vectorized because a recurrence was found on "DATA" at line 59.

ftn-6751 f90: STREAM File = meas.F, Line = 45
A loop starting at line 45 was not multi-streamed because a recurrence was found on "DATA" at line 59.

Problem loops (meas.F)

```
45.  1---------------<        do ic=1,Nc
46.  1 2-------------<         do jc=1,Nc
47.  1 2 3-----------<          do n=1,nl
48.  1 2 3 Vp--------<            do m=1,nl
49.  1 2 3 Vp                      k=n-m
50.  1 2 3 Vp                      uno=halfr
51.  1 2 3 Vp                      if(k.lt.0) then
52.  1 2 3 Vp                        uno=-halfr
53.  1 2 3 Vp                        k=k+nl
54.  1 2 3 Vp                      end if
55.  1 2 3 Vp                      k=k+1
56.  1 2 3 Vp                      i=icrdiff(ic,jc)
57.  1 2 3 Vp                      ind=ic + (n-1)*Nc
58.  1 2 3 Vp                      imd=jc + (m-1)*Nc
59.  1 2 3 Vp r-----<>                data(k,i)=data(k,i)+(gu(ind,imd)+gd(ind,imd))*uno
60.  1 2 3 Vp-------->            end do
61.  1 2 3----------->          end do
62.  1 2------------->         end do
63.  1--------------->        end do
```

Further analysis shows…

- data depends on i and k
- k depends on m and n
  - For fixed n, k is unique across m values
  - m loop is “concurrent”
- i depends on icrdiff(ic,jc)
  - For fixed ic, i is unique across jc values
  - jc loop is “concurrent”

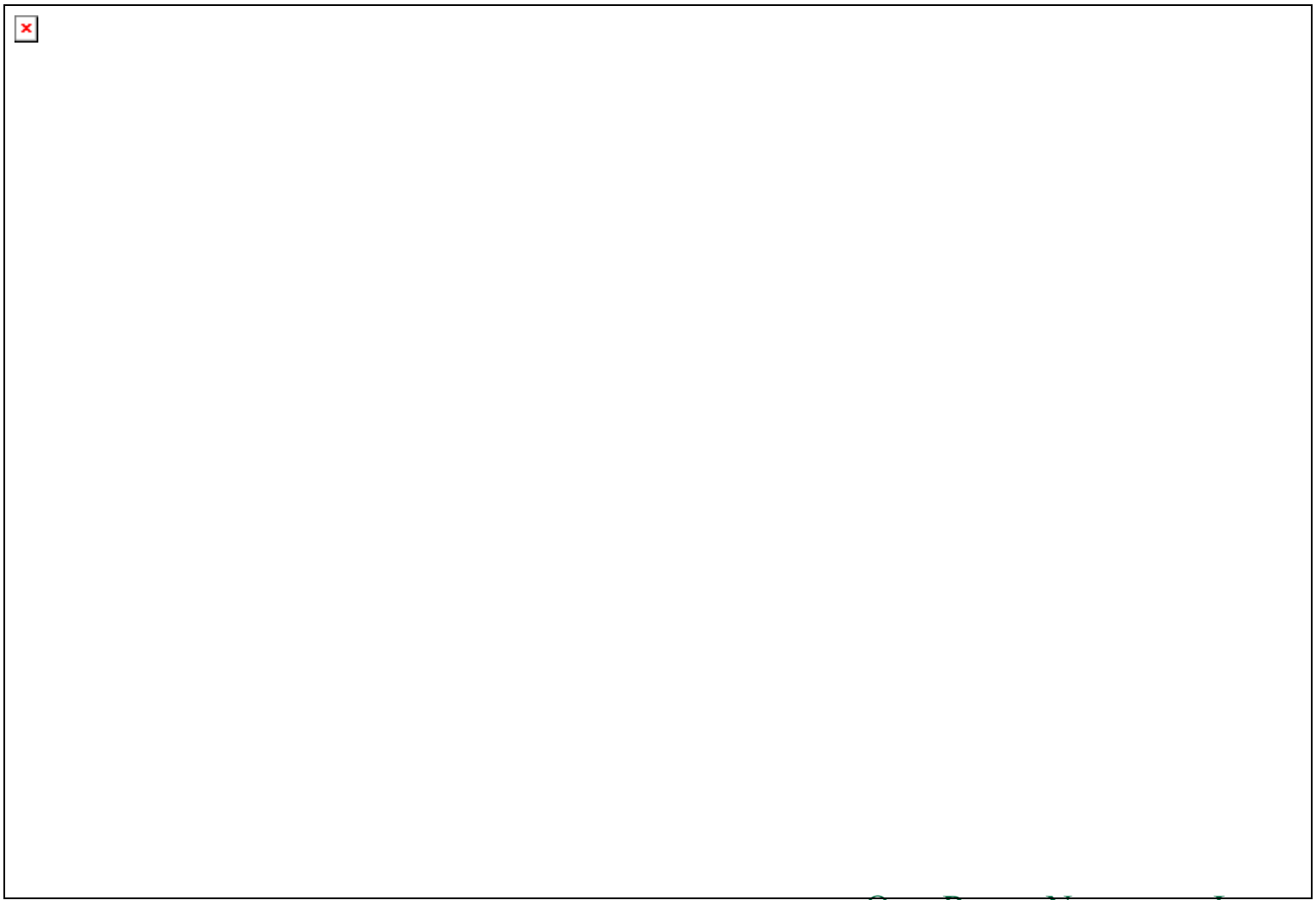
Success!

```
45.  1------------<        do ic=1,Nc
46.  1             cdir$ concurrent
47.  1 M----------<        do jc=1,Nc
48.  1 M 3--------<        do n=1,nl
49.  1 M 3         cdir$ concurrent
50.  1 M 3 V------<        do m=1,nl
51.  1 M 3 V                 k=n-m
52.  1 M 3 V                 uno=halfr
53.  1 M 3 V                 if(k.lt.0) then
54.  1 M 3 V                   uno=-halfr
55.  1 M 3 V                   k=k+nl
56.  1 M 3 V                 end if
57.  1 M 3 V                 k=k+1
58.  1 M 3 V                 i=icrdiff(ic,jc)
59.  1 M 3 V                 ind=ic + (n-1)*Nc
60.  1 M 3 V                 imd=jc + (m-1)*Nc
61.  1 M 3 V                 data(k,i)=data(k,i)+(gu(ind,imd)+gd(ind,imd))*uno
62.  1 M 3 V------>        end do
63.  1 M 3-------->        end do
64.  1 M---------->        end do
65.  1------------>        end do
```

DCA-QMC test cases

- Small
  - 4 atoms, 10 time values
  - 20 warm up, 4 skipped, 320 measurements
- Medium
  - 16 atoms, 16 time values
  - 20 warm up, 4 skipped, 320 measurements
- Large
  - 64 particles, 16 time values
  - 1 warm up, 4 skipped, 1 measurement
- Preproduction X1 (700 MHz instead of 800 MHz)

DCA-QMC Runtime



Cray X1 profile for 64x16x1

- 23.3% 23.3% 3388 _sd2uge
- 9.9% 33.2% 1433 _ld_write
- 8.0% 41.2% 1166 __write
- 7.8% 49.1% 1135 %__cabs
- 6.8% 55.9% 991 _beautify
- 6.1% 62.0% 890 %__csqrt
- 5.1% 67.1% 738 perturbation_
- …

Cray X1 profile for 64x16x1

- 23.3% 23.3% 3388 _sd2uge
- 9.9% 33.2% 1433 _ld_write
- 8.0% 41.2% 1166 __write
- 7.8% 49.1% 1135 %__cabs
- 6.8% 55.9% 991 _beautify
- 6.1% 62.0% 890 %__csqrt
- 5.1% 67.1% 738 perturbation_
- …

Vectorized functions

Cray X1 profile for 64x16x1

- 23.3% 23.3% 3388 _sd2uge
- 9.9% 33.2% 1433 _ld_write
- 8.0% 41.2% 1166 __write
- 7.8% 49.1% 1135 %__cabs
- 6.8% 55.9% 991 _beautify
- 6.1% 62.0% 890 %__csqrt
- 5.1% 67.1% 738 perturbation_
- …

???

Cray X1 profile for 64x16x1

- 23.3% 23.3% 3388 _sd2uge
- 9.9% 33.2% 1433 _ld_write
- 8.0% 41.2% 1166 __write
- 7.8% 49.1% 1135 %__cabs
- 6.8% 55.9% 991 _beautify
- 6.1% 62.0% 890 %__csqrt
- 5.1% 67.1% 738 perturbation_
- …

Formatted Fortran I/O

Performance summary

- Minimal code modifications
  - Directives/comments added to one file
- 700 MHz X1 shows 7.4x performance of p690
- Half X1 runtime in formatted output
  - Negligible on p690
- Further optimization
  - Bulk, unformatted (binary) output?
  - Optimize **`cabs`** and **`csqrt`** (Cray)
  - Re-arrange some loops
  - Vectorize and multistream remaining loops