

計算の概略

QCD のシミュレーションにおいて、BlueGene で大きなゲージ配位を書き出す。

使用言語

Fortran

問題

高速化したい。配位が大きい ($32^3 \times 64$) の時、IO に非常に時間がかかって困っている。
他の格子サイズのものと同じ型の出力ファイルにしたいので、中身を変えることは避けたい。
多分、MPI をうまく使ってやる方法があると思うのですがわからないので、具体的に書き方を教えて欲しい。

解決法

高速化したい部分は以下のサブルーチンである。

```
subroutine Ufin(Iconf)
  use lattsize
  use hmccomm
!-----Variables-----
  character(len=120) WFILE

!  COMMON /Link_tmp/  Utmp(Ndf,Nszx*Nszy*Nszz*Ntime,4)
!  COMMON /Link_tmp2/ Utmp2(Ndf,Nszx*Nszy*Nszz,4,Ntime)

  double precision UW(Ndf,Nszx*Nszy*Nszz,4)
  double precision Utmp(Ndf,Nszx*Nszy*Nszz,4)

  double precision Utm(Ndf,Nst,4)

  integer idir, is, ist, idf
  integer ix, iy, iz, it

  integer ipe2
  integer ix2, iy2, iz2, it2, ist2
  integer ipex, ipey, ipez, ipet

!-----Main-----
```

```

! >>>

do ipet = 0, NPet - 1
  do it = 0, Nt - 1

    ! initialize
    UW(:, :, :) = 0.0d0 ! Ndf, Ns, 4

    do ipex = 0, NPEX - 1; do ipey = 0, NPEY - 1; do izez = 0, NPEZ - 1
      ipe2 = ipex + NPEX*ipey + NPEX*NPEY*izez + NPEX*NPEY*NPEZ*ipet

      if(ipe2.eq.0) then
        call Equate( Ndf*4*Nst, Utm, U )
      else
        call Send_lto1( Ndf*4*Nst, Utm, U, 0, ipe2, ipe2 )
      endif

      ! previous data
      call Equate( Ndf*4*Nsrx*Nszy*Nszz, Utmp, UW )

      if (NODEID.eq.0) then
        do iz = 0, Nz - 1
          iz2 = iz + izez*Nz
          do iy = 0, Ny - 1
            iy2 = iy + ipey*Ny
            do ix = 0, Nx - 1
              ist = 1 + ix + iy*Nx + iz*Nx*Ny + it*Nx*Ny*Nz ! sub-lattice index

              ix2 = ix + ipex*Nx
              is2 = 1 + ix2 + iy2*Nsrx + iz2*Nsrx*Nszy ! full-lattice index w/o time

              do idir = 1, 4; do idf = 1, Ndf
                Utmp(idf, is2, idir) = Utm(idf, ist, idir)
              enddo; enddo
            enddo
          enddo
        enddo ! iz
      endif ! NODEID = 0
    enddo
  enddo
enddo

```

```

! update sub-lattice data
call Equate( Ndf*4*Nsxx*Nszy*Nszz, UW, Utmp )

enddo; enddo; enddo; ! ipex,ipey,ipez

call SyncPE

itime = it + Nt*ipet ! full-lattice time index
if (NODEID .eq. 0) then
  write(WFILE, '(("./ConfU/", i4.4, "/G",i4.4,".", i2.2)') Iconf,Iconf, itime
  !---org write(WFILE, '(("./ConfU/G", i4.4, ".", i2.2)') Iconf, itime
  write(*,*) ' write ', WFILE

  open(UNIT=12, FILE=WFILE,STATUS='UNKNOWN', FORM='UNFORMATTED')
  write(12) UW
  close(12)
endif

enddo ! it-loop
enddo ! ipet-loop

end subroutine Ufin

```

担当者の改良版をもとに依頼者が次のような版をつくった。

```

! do ipet = 0, NPEt - 1
  do it = 0, Nt - 1

    ! initialize
    UW(:, :, :) = 0.0d0 ! Ndf, Ns, 4

    do ipex = 0, NPEx - 1; do ipey = 0, NPEy - 1; do ipez = 0, NPEz - 1
!       ipe2 = ipex + NPEx*ipey + NPEx*NPEy*ipez + NPEx*NPEy*NPEz*ipet
       ipe2 = ipex + NPEx*ipey + NPEx*NPEy*ipez + NPEx*NPEy*NPEz*IPE(4)

!       if(ipe2.eq.0) then

```

```

if (mod(ipe2,NPEx*NPEy*NPEz).eq.0) then
  call Equate( Ndf*4*Nst, Utm, U )
else
!   call Send_1tol( Ndf*4*Nst, Utm, U, 0, ipe2, ipe2 )
!   Call Send_1tol( Ndf*4*Nst, Utm, U, NPEx*NPEy*NPEz*ipet, ipe2, ipe2)
      Call Send_1tol( Ndf*4*Nst, Utm, U, NPEx*NPEy*NPEz*IPe(4), ipe2, ipe2)

endif

! previous data
call Equate( Ndf*4*Nszx*Nszy*Nszz, Utmp, UW )

!   if (NODEID.eq.0) then
if (mod(nodeid, NPEx*NPEy*NPEz).eq.0) then
  do iz = 0, Nz - 1
    iz2 = iz + ipez*Nz
    do iy = 0, Ny - 1
      iy2 = iy + ipey*Ny
      do ix = 0, Nx - 1
        ist = 1 + ix + iy*Nx + iz*Nx*Ny + it*Nx*Ny*Nz ! sub-lattice index

        ix2 = ix + ipex*Nx
        is2 = 1 + ix2 + iy2*Nszx + iz2*Nszx*Nszy ! full-lattice index w/o time

        do idir = 1, 4; do idf = 1, Ndf
          Utmp(idf,is2,idir) = Utm(idf,ist,idir)
        enddo; enddo
      enddo
    enddo
  enddo ! iz
endif ! NODEID = 0

! update sub-lattice data
call Equate( Ndf*4*Nszx*Nszy*Nszz, UW, Utmp )

enddo; enddo; enddo; ! ipex,ipey,ipez

call SyncPE

```

```

!           itime = it + Nt*ipet ! full-lattice time index
           itime = it + Nt*IPE(4) ! full-lattice time index
!
! if (NODEID .eq. 0) then
!   if (mod(nodeid,NPEx*NPEy*NPEz).eq.0) then
!     write(WFILE, '(("./ConfU/", i4.4, "/G",i4.4,".", i2.2)') Iconf,Iconf, itime
!     !---org write(WFILE, '(("./ConfU/G", i4.4, ".", i2.2)') Iconf, itime
!     write(*,*) ' write ', WFILE
!
!     open(UNIT=12, FILE=WFILE,STATUS='UNKNOWN', FORM='UNFORMATTED')
!     open(UNIT=mod(nodeid,NPEx*NPEy*NPEz), FILE=WFILE,STATUS='UNKNOWN', FORM='UNFORMATTED')
!     open(UNIT=mod(nodeid,NPEx*NPEy*NPEz)+7, FILE=WFILE,STATUS='UNKNOWN', FORM='UNFORMATTED')
!     write(12) UW
!     write(mod(nodeid,NPEx*NPEy*NPEz)) UW
!     write(mod(nodeid,NPEx*NPEy*NPEz)+7) UW
!     close(12)
!     close(mod(nodeid,NPEx*NPEy*NPEz))
!     close(mod(nodeid,NPEx*NPEy*NPEz)+7)
!
!   endif
!
!   enddo ! it-loop
!   enddo ! ipet-loop

```

要点は、ipet ループをなくし、t 方向のノード番号をグローバル変数 `ipe(4)` にしたことと、今まで 0 番プロセスが書き出しをしていたのを、プロセス番号 `NPEx*NPEy*NPEz*IPE(4)` のプロセスがデータをファイルへ書き出すようにしたことである。これにより、これまで配位の書き出しに 3 時間 10 分ほどかかっていたのが、6 分で終わるようになった。