Introduction to Parallel Programming for Multicore/Manycore Clusters

Part VI: Report

Kengo Nakajima
Information Technology Center

Programming Exercise

- Target: **`multicore/omp/src0`**
  - http://nkl.cc.u-tokyo.ac.jp/15s/OpenMP.pdf
  - PCG with Diagonal Scaling Preconditioning
  - Apply “ELL” matrix storage format to the target

- Comparison of performance
- Analysis by profiler

OpenMP Version (C)

```
>$ cd
>$ cp /home/z30088/omp/omp-c0.tar .
>$ tar xvf omp-c0.tar
>$ cd multicore/omp/src0
>$ make

>$ ls ../run/sol0
   sol0

(modify go.sh (see below))
>$ pjsub go.sh
```

```
#!/bin/sh
#PJM -L "node=1"
#PJM -L "elapse=00:10:00"
#PJM -L "rscgrp=lecture9"
#PJM -g "gt89"
#PJM -j
#PJM -o “aaa.lst"
export OMP_NUM_THREADS=M      M=1,4 8,16

./sol0
```

OpenMP Version (Fortran)

```
>$ cd
>$ cp /home/z30088/omp/omp-f0.tar .
>$ tar xvf omp-f0.tar
>$ cd multicore/omp/src0
>$ make

>$ ls ../run/sol0
   sol0

(modify go.sh (see below))
>$ pjsub go.sh
```

```
#!/bin/sh
#PJM -L "node=1"
#PJM -L "elapse=00:10:00"
#PJM -L "rscgrp=lecture9"
#PJM -g "gt89"
#PJM -j
#PJM -o “aaa.lst"
export OMP_NUM_THREADS=M      M=1,4 8,16

./sol0
```

Original Program

```
>$ cd <$O-TOP>

>$ cp /home/z30088/class_eps/ex.tar .

>$ tar xvf ex.tar

>$ cd ex

>$ cd C
>$ ls
     run    src

>$ cd ../F
>$ ls
     run    src

<$O-TOP>/F,<$O-TOP>/C: <$O-ex>
```

Procedures

- Code is parallelized, and redordered
  - OpenMP
  - Reordered CM/RCM
- Apply ELL
  - poi_gen
  - solver_PCG

Example: poi_gen

For all rows, number of non-zero off-diagonal components= 6（=3+3）

```
!C
!C-- 1D array

      allocate (indexL(0:nn), indexU(0:nn))
      indexL= 0
      indexU= 0

      do icel= 1, ICELTOT
        indexL(icel)= 3
        indexU(icel)= 3
      enddo

      do icel= 1, ICELTOT
        indexL(icel)= indexL(icel) + indexL(icel-1)
        indexU(icel)= indexU(icel) + indexU(icel-1)
      enddo

      NPL= indexL(ICELTOT)
      NPU= indexU(ICELTOT)

      allocate (itemL(NPL), AL(NPL))
      allocate (itemU(NPU), AU(NPU))

      itemL= 0
      itemU= 0
         AL= 0.d0
         AU= 0.d0
```

Example: poi_gen

Column ID if AL/AU= 0.0: ICELTOT+icou

```
      icou= 0
      do i= 1, ICELTOT
      do k= 1, 3
        if (itemL(indexL(i-1)+k).eq.0) then
          icou= icou + 1
          itemL(indexL(i-1)+k)= ICELTOT + icou
        endif
        if (icou.eq.N4) icou= 0
      enddo
      enddo

      icou= 0
      do i= 1, ICELTOT
      do k= 1, 3
        if (itemU(indexU(i-1)+k).eq.0) then
          icou= icou + 1
          itemU(indexU(i-1)+k)= ICELTOT + icou
        endif
        if (icou.eq.N4) icou= 0
      enddo
      enddo
```

```
icou: 1-N4
N4: Not so big value (e.g.256)

PHI(ICELTOT+N4)
```

Example: solver_PCG

Implementation of Mat-Vec (1/3)

```
      do i= 1, N
        VAL= D(i)*W(i,P)

        do k= indexL(i-1)+1, indexL(i-1)+3
          VAL= VAL + AL(k)*W(itemL(k),P)
        enddo

        do k= indexU(i-1)+1, indexU(i-1)+3
          VAL= VAL + AU(k)*W(itemU(k),P)
        enddo

        W(i,Q)= VAL
      enddo
```

```
W(ICELTOT+N4,4)
```

Example: solver_PCG

Implementation of Mat-Vec (2/3)

```
      do i= 1, N
        VAL= D(i)*W(i,P)

        do k= 1, 3
          kk= (i-1)*3 + k
          VAL= VAL + AL(kk)*W(itemL(kk),P)
        enddo

        do k= 1, 3
          kk= (i-1)*3 + k
          VAL= VAL + AU(kk)*W(itemU(kk),P)
        enddo

        W(i,Q)= VAL
      enddo
```

```
W(ICELTOT+N4,4)
```

Example: solver_PCG

Implementation of Mat-Vec (3/3)

```
      do i= 1, N
        VAL= D(i)*W(i,P)

        do k= 1, 3
          kk= (i-1)*3 + k
          VAL= VAL + AL(kk)*W(itemL(kk),P)
     &                 + AU(kk)*W(itemU(kk),P)
        enddo

        W(i,Q)= VAL
      enddo
```

```
      do i= 1, N
        VAL= D(i)*W(i,P)

        do k= 1, 6
          kk= (i-1)*6 + k
          VAL= VAL + AMAT(kk)*W(itemLU(kk),P)
        enddo

        W(i,Q)= VAL
      enddo
```

Report

- Deadline: 17:00 August 29th (Sat), 2015.
  - Send files via e-mail at **`nakajima(at)cc.u-tokyo.ac.jp`**
- Report
  - Cover Page: Name, ID, and Problem ID (P1) must be written.
  - Less than <u>20</u> pages including figures and tables (A4).
    - Strategy
    - Structure of the Program
    - Numerical Experiments
    - Performance Analysis by Profiler
    - Remarks
  - <u>Source list of the program</u>