Parallel Programming for Multicore Processors using OpenMP

Part IV: Report P1

Kengo Nakajima
Information Technology Center

Programming for Parallel Computing (616-2057)
Seminar on Advanced Computing (616-4009)

Programming Exercise

- Target: “L1-sol with METHOD=3” (Diagonal Scaling)
  - Develop a parallel version of the target using OpenMP
  - Apply “ELL” matrix storage format to the target

- Comparison of performance
- Analysis by profiler

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>
```

Files on FX10

- Location
  - `<$O-ex>/src`, `<$O-ex>/run`
- Compile/Run
  - Main Part
    - `cd <$O-ex>/src`
    - `make`
    - `<$O-ex>/run/L1-sol` (exec)
  - Control Data
    - `<$O-ex>/run/INPUT.DAT`
  - Batch Job Script
    - Please make it by yourself

Procedures

- Code is not parallelized
  - based on METHOD=3 of L1-sol
- Apply OpenMP
  - You do not have to use “SMPindexG”
- 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.N2) 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.N2) icou= 0
      enddo
      enddo
```

```
icou: 1-N2
N2: Not so big value (e.g.256)

PHI(ICELTOT+N2)
```

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+N2,4)
```

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+N2,4)
```

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 P1

- Deadline: 17:00 October 12th (Sat), 2013.
  - 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 20 pages including figures and tables (A4).
    - Strategy
    - Structure of the Program
    - Numerical Experiments, Performance Analysis
    - Remarks
  - Source list of the program
- Grade
  - A（優）might be given, if “ELL” is done.
  - B（良）is the highest grade if “ELL” is NOT done.
    - Even if you have not done “ELL”, please submit your report !

Grading by Reports ONLY

- MPI (Collective Communication) (S1)
- MPI (1D Parallel FEM) (S2)
- Parallel FEM (S3)
  - If you complete (S1-S3), you get credits of “Programming for Parallel Computing (616-2057)” .
- OpenMP (P1)
  - If you complete (P1), you get credits of “Seminar on Advanced Computing (616-4009)” are graded.

- Sample solutions will be available (S1, S2)
- Deadline: October 12th (Sat) 17:00
  - By E-mail: nakajima(at)cc.u-tokyo.ac.jp
  - You can bring hard-copy’s to my office ...