Introduction to Parallel Programming for Multicore/Manycore Clusters

Report

Kengo Nakajima
Information Technology Center
The University of Tokyo

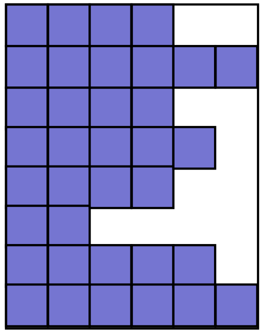
Going back to Part-A2

- Parallel PCG by OpenMP
- src-c2, src-f2: ELL
- src-c3, src-f3: reduced “omp parallel”

Storage of Sparse Matrices (C)

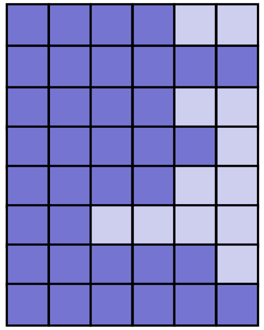
CRS（Compressed Row Storage）

```
for(i=0; i<N; i++) {
      VAL = D[i] * W[P][i];
      for(j=indexLU[i]; j<indexLU[i+1]; j++) {
           VAL += AMAT[j] * W[P][itemLU[j]];
      }
      W[Q][i] = VAL;}
```



ELL (ELLPACK/ITPACK)

```
for(i=0; i<N; i++) {
      VAL = D[i] * W[P][i];
      for(j=0; j<6; j++) {
           VAL += AMAT[6*i+j] * W[P][itemLU[6*i+j]];
      }
      W[Q][i] = VAL;}
```



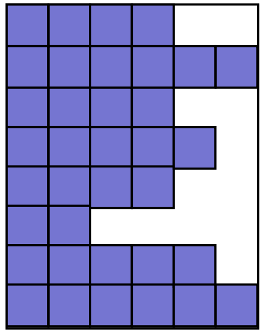
ELL (ELLPACK/ITPACK)

```
for(i=0; i<N; i++) {
      VAL = D[i] * W[P][i];
      for(j=0; j<6; j++) {
           VAL += AMAT[i][j] * W[P][itemLU[i][j]];
      }
      W[Q][i] = VAL;}
```

Storage of Sparse Matrices (F)

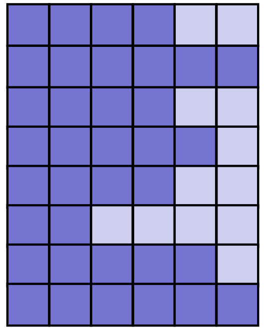
CRS（Compressed Row Storage）

```
do i= 1, N
  W(i,Q)= D(i)*W(i,P)
  do k= indexLU(i-1)+1, indexLU(i)
    W(i,Q)= W(i,Q) + AMAT(k)*W(itemLU(k),P)
  enddo
enddo
```



ELL (ELLPACK/ITPACK)

```
do i= 1, N
  W(i,Q)= D(i)*W(i,P)
  do j= 1, 6
    W(i,Q)= W(i,Q) + AMAT(j,i)*W(itemLU(j,i),P)
  enddo
enddo
```

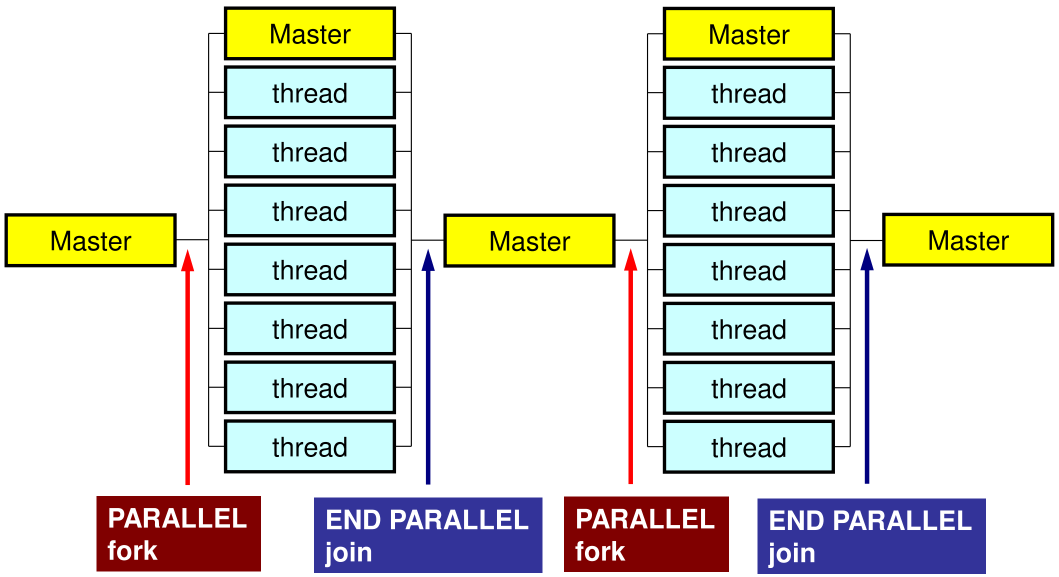


ELL (ELLPACK/ITPACK)

```
do i= 1, N
  W(i,Q)= D(i)*W(i,P)
  do j= 1, 6
    W(i,Q)= W(i,Q) + AMAT(6*(i-1)+k)*W(itemLU(6*(i-1)+j),P)
  enddo
enddo
```

omp parallel (do)

- "omp parallel-omp end parallel" = "fork-join"
- If you have many loops, these "fork-join's" cause overheads
- omp parallel + omp do/omp for




```
#pragma omp parallel ...

#pragma omp for {


...
#pragma omp for {
```

```
!$omp parallel ...

!$omp do
   do i= 1, N
...
!$omp do
   do i= 1, N
...
!$omp end parallel   required
```

!$omp parallel do: Fork-Join (C)

```
#pragma omp parallel for private (i,VAL,j)
  for(i=0; i<N; i++) {
      VAL = D[i] * W[P][i];
      for(j=indexLU[i]; j<indexLU[i+1]; j++) {
          VAL += AMAT[j] * W[P][itemLU[j]];
      }
      W[Q][i] = VAL;
  }

C1 = 0.0;
#pragma omp parallel for private (i) reduction(+:C1)
  for(i=0; i<N; i++) {
      C1 += W[P][i] * W[Q][i];
  }
  ALPHA = RHO / C1;

#pragma omp parallel for private (i)
  for(i=0; i<N; i++) {
      X[i]    += ALPHA * W[P][i];
      W[R][i] -= ALPHA * W[Q][i];
  }

DNRM2 = 0.0;
#pragma omp parallel for private (i) reduction(+:DNRM2)
  for(i=0; i<N; i++) {
      DNRM2 += W[R][i]*W[R][i];
  }

  ERR = sqrt(DNRM2/BNRM2);
...
```

Compute $r^{(0)}= b-[A]x^{(0)}$

for $i= 1, 2, \ldots$

  solve $[M]z^{(i-1)}= r^{(i-1)}$

  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$

  if $i=1$

    $p^{(1)}= z^{(0)}$

  else

    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$

    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i-1)}$

  endif

  $\mathbf{q}^{(i)}= [\mathbf{A}]\mathbf{p}^{(i)}$

  $\alpha_i = \rho_{i-1}/\mathbf{p}^{(i)}\mathbf{q}^{(i)}$

  $\mathbf{x}^{(i)}= \mathbf{x}^{(i-1)} + \alpha_i\mathbf{p}^{(i)}$

  $\mathbf{r}^{(i)}= \mathbf{r}^{(i-1)} - \alpha_i\mathbf{q}^{(i)}$

  check convergence $|\mathbf{r}|$

end

!$omp parallel do: Fork-Join (F)

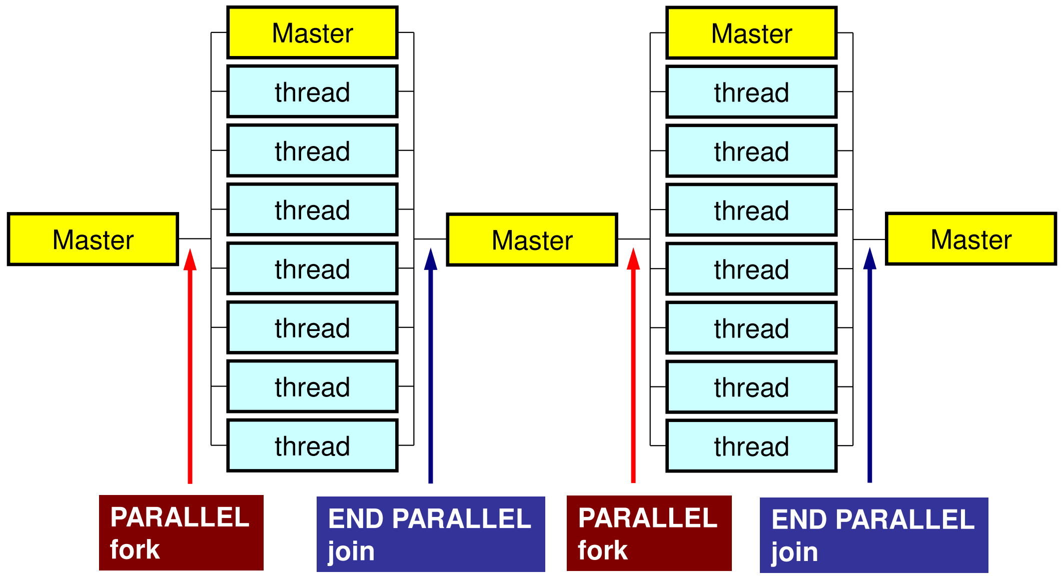
```
!$omp parallel do private(i,VAL,k)
      do i= 1, N
        VAL= D(i)*W(i,P)
        do k= indexLU(i-1)+1, indexLU(i)
          VAL= VAL + AMAT(k)*W(itemLU(k),P)
        enddo
        W(i,Q)= VAL
      enddo

      C1= 0.d0
!$omp parallel do private(i) reduction(+:C1)
      do i= 1, N
        C1= C1 + W(i,P)*W(i,Q)
      enddo

      ALPHA= RHO / C1

!$omp parallel do private(i)
      do i= 1, N
        X(i)  = X(i)   + ALPHA * W(i,P)
        W(i,R)= W(i,R) - ALPHA * W(i,Q)
      enddo

      DNRM2= 0.d0
!$omp parallel do private(i) reduction(+:DNRM2)
      do i= 1, N
        DNRM2= DNRM2 + W(i,R)**2
      enddo

      ERR = dsqrt(DNRM2/BNRM2)...
```

Compute $r^{(0)}= b-[A]x^{(0)}$
<u>for</u> i= 1, 2, …
- solve $[M]z^{(i-1)}= r^{(i-1)}$
- $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
- <u>if</u> i=1
  - $p^{(1)}= z^{(0)}$
- <u>else</u>
  - $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
  - $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i-1)}$
- <u>endif</u>
- $\mathbf{q}^{(i)}= [\mathbf{A}]\mathbf{p}^{(i)}$
- $\alpha_i = \rho_{i-1}/\mathbf{p}^{(i)}\mathbf{q}^{(i)}$
- $\mathbf{x}^{(i)}= \mathbf{x}^{(i-1)} + \alpha_i\mathbf{p}^{(i)}$
- $\mathbf{r}^{(i)}= \mathbf{r}^{(i-1)} - \alpha_i\mathbf{q}^{(i)}$
- **check convergence** $|\mathbf{r}|$

<u>end</u>

Strategy for Further Optimization

- src-c3, src-f3
  - Only 1 omp-parallel in each iteration



src_f3 (1/2)

```
      ITR= N
      Stime= omp_get_wtime()

      do L= 1, ITR
!$omp parallel private(i,k,VAL)

!$omp do
      do i= 1, N
        W(i,Z)= W(i,R)*W(i,DD)
      enddo

      RHO= 0.d0
!$omp do reduction(+:RHO)
      do i= 1, N
        RHO= RHO + W(i,R)*W(i,Z)
      enddo

      if ( L.eq.1 ) then
!$omp do
        do i= 1, N
          W(i,P)= W(i,Z)
        enddo
       else
         BETA= RHO / RHO1
!$omp do
         do i= 1, N
           W(i,P)= W(i,Z) + BETA*W(i,P)
         enddo
      endif
```

Compute $r^{(0)}= b-[A]x^{(0)}$

for i= 1, 2, …

  solve $[M]z^{(i-1)}= r^{(i-1)}$

  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$

  if i=1

   $p^{(1)}= z^{(0)}$

  else

   $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$

   $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i-1)}$

  endif

  $q^{(i)}= [A]p^{(i)}$

  $\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$

  $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$

  $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$

  check convergence $|r|$

end

src_f3 (2/2)

```
!$omp do
      do i= 1, N
        VAL= D(i)*W(i,P)
        do k= indexLU(i-1)+1, indexLU(i)
          VAL= VAL + AMAT(k)*W(itemLU(k),P)
        enddo
        W(i,Q)= VAL
      enddo

      C1= 0.d0
!$omp do reduction(+:C1)
      do i= 1, N
        C1= C1 + W(i,P)*W(i,Q)
      enddo

      ALPHA= RHO / C1

!$omp do
      do i= 1, N
        X(i)  = X(i)   + ALPHA * W(i,P)
        W(i,R)= W(i,R) - ALPHA * W(i,Q)
      enddo

      DNRM2= 0.d0
!$omp do reduction(+:DNRM2)
      do i= 1, N
        DNRM2= DNRM2 + W(i,R)**2
      enddo

!$omp end parallel
      ERR = dsqrt(DNRM2/BNRM2)...
```

Compute $r^{(0)}= b-[A]x^{(0)}$
<u>for</u> i= 1, 2, …
    solve $[M]z^{(i-1)}= r^{(i-1)}$
    $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
    <u>if</u> i=1
        $p^{(1)}= z^{(0)}$
    <u>else</u>
        $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
        $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i-1)}$
    <u>endif</u>
    $\mathbf{q^{(i)}= [A]p^{(i)}}$
    $\boldsymbol{\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}}$
    $\mathbf{x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}}$
    $\mathbf{r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}}$
    check convergence |r|
<u>end</u>

src_c3 (1/2)

```
 *ITR = N;

  Stime = omp_get_wtime();
  for(L=0; L<(*ITR); L++) {

#pragma omp parallel private (i,j,VAL){

#pragma omp for
    for(i=0; i<N; i++) {
      W[Z][i] = W[R][i]*W[DD][i];
    }

    RHO = 0.0;
#pragma omp for reduction(+:RHO)
    for(i=0; i<N; i++) {
      RHO += W[R][i] * W[Z][i];
    }

if(L == 0) {
#pragma omp for
    for(i=0; i<N; i++) {
      W[P][i] = W[Z][i];
    }
    } else {
     BETA = RHO / RHO1;
#pragma omp for
    for(i=0; i<N; i++) {
      W[P][i] = W[Z][i] + BETA * W[P][i];
    }
}
```

Compute $r^{(0)}= b-[A]x^{(0)}$
<u>for</u> i= 1, 2, …
    solve $[M]z^{(i-1)}= r^{(i-1)}$
    $\rho_{i-1}= r^{(i-1)} z^{(i-1)}$
    <u>if</u> i=1
      $p^{(1)}= z^{(0)}$
     <u>else</u>
      $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
      $p^{(i)}= z^{(i-1)} + \beta_{i-1} p^{(i-1)}$
    <u>endif</u>
    $q^{(i)}= [A]p^{(i)}$
    $\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}$
    $x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}$
    $r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}$
    check convergence $|r|$
<u>end</u>

src_c3 (2/2)

```
#pragma omp for
  for(i=0; i<N; i++) {
      VAL = D[i] * W[P][i];
      for(i=0; i<6; i++) {
          VAL += AMAT[6*i+j] * W[P][itemLU[6*i+j]];
      }
      W[Q][i] = VAL;
  }

  C1 = 0.0;
#pragma omp for reduction(+:C1)
  for(i=0; i<N; i++) {
      C1 += W[P][i] * W[Q][i];
  }
  ALPHA = RHO / C1;

#pragma omp for
  for(i=0; i<N; i++) {
      X[i]    += ALPHA * W[P][i];
      W[R][i] -= ALPHA * W[Q][i];
  }

  DNRM2 = 0.0;
#pragma omp for reduction(+:DNRM2)
  for(i=0; i<N; i++) {
      DNRM2 += W[R][i]*W[R][i];
  }

}

ERR = sqrt(DNRM2/BNRM2);
```

Compute $r^{(0)}= b-[A]x^{(0)}$
<u>for</u> $i= 1, 2, \dots$
  solve $[M]z^{(i-1)}= r^{(i-1)}$
  $\rho_{i-1}= r^{(i-1)}\ z^{(i-1)}$
  <u>if</u> $i=1$
    $p^{(1)}= z^{(0)}$
  <u>else</u>
    $\beta_{i-1}= \rho_{i-1}/\rho_{i-2}$
    $p^{(i)}= z^{(i-1)} + \beta_{i-1}\ p^{(i-1)}$
  <u>endif</u>
  $\mathbf{q^{(i)}= [A]p^{(i)}}$
  $\boldsymbol{\alpha_i = \rho_{i-1}/p^{(i)}q^{(i)}}$
  $\mathbf{x^{(i)}= x^{(i-1)} + \alpha_i p^{(i)}}$
  $\mathbf{r^{(i)}= r^{(i-1)} - \alpha_i q^{(i)}}$
  check convergence $|r|$
<u>end</u>

Programming Exercise for Report

- Implement the following items to “reorder0 (L3-rsol0)” with sequential reordering
  - ELL (src-c2/src-f2)
  - Reduced “omp parallel” in (src-c3/src-f3)
- CM-RCM(2)
- Evaluations using 48 threads

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

→

11	2	9	1
5	12	3	10
15	6	13	4
8	16	7	14

CM-RCM(2)

→

7	2	5	1
9	8	3	6
15	10	13	4
12	16	11	14

2 threads, sequential
1-8: #0 thread
9-16: #1 thread

Report

- <u>**Deadline: 17:00 August 17 (W), 2022**</u>
- <u>**Upload the Report at ITC-LMS**</u>
- **Report：日本語でもOK**
  - **Cover Page: Name and ID must be written.**
  - **No more than <u>20</u> pages including figures and tables (A4).**
    - **Strategy**
    - **Structure of the Program, Details of Modification**
    - **Validation**
    - **Numerical Experiments**
    - **Performance Analysis, Detailed Profiler**
    - **Remarks (Important !!)**
  - <u>**Source list of the entire program (not included in the 20 pages above)**</u>

Forward/Backward Substitution solve Mz=r, ELL

ic=1: Only Upper Part (AU)
ic=2: Only Lower Part (AL)

```
!$omp parallel do private(i)
      do i= 1, N
        W(i,Z)= W(i,R)
      enddo

!$omp parallel private(ic,ip,ip1,i,WVAL,k)

      ic= 1
!$omp do
      do ip= 1, PEsmpTOT
        ip1= (ip-1)*NCOLORtot + ic
        do i= SMPi(ip1-1)+1, SMPi(ip1)
          W(i,Z)= W(i,Z) * W(i,DD)
        enddo
      enddo
      enddo

      ic= NCOLORtot
!$omp do
      do ip= 1, PEsmpTOT
        ip1= (ip-1)*NCOLORtot + ic
        do i= SMPi(ip1-1)+1, SMPi(ip1)
          WVAL= W(i,Z)
          do k= 1, 6
            WVAL= WVAL -  AL(k,i) * W(itemL(k,i),Z)
          enddo
          W(i,Z)= WVAL * W(i,DD)
        enddo
      enddo
!$omp end parallel
```

```
!$omp parallel private(ic,ip,ip1,i,SW,k)

      ic= 1
!$omp do
      do ip= 1, PEsmpTOT
        ip1= (ip-1)*NCOLORtot + ic
        do i= SMPi(ip1-1)+1, SMPi(ip1)
          SW  = 0.0d0
          do k= 1, 6
            SW= SW + AU(k,i) * W(itemU(k,i),Z)
          enddo
          W(i,Z)= W(i,Z) - W(i,DD) * SW
        enddo
      enddo

!$omp end parallel
```

SpMV: Sparse Mat. Vec. Multiply
Ap= q, ELL
ic=1: Only Upper Part (AU)
ic=2: Only Lower Part (AL)

```
!$omp parallel private(ic,ip,ip1,i,VAL,k)

      ic= 1
!$omp do
      do ip= 1, PEsmpTOT
        ip1= (ip-1)*NCOLORtot + ic
      do i= SMPi(ip1-1)+1, SMPi(ip1)
        VAL= D(i)*W(i,P)
        do k= 1, 6
          VAL= VAL +
     &         AU(k,i)*W(itemU(k,i),P)
        enddo
        W(i,Q)= VAL
      enddo
      enddo
```

```
      ic= NCOLORtot
!$omp do
      do ip= 1, PEsmpTOT
        ip1= (ip-1)*NCOLORtot + ic
      do i= SMPi(ip1-1)+1, SMPi(ip1)
        VAL= D(i)*W(i,P)
        do k= 1, 6
          VAL= VAL +
     &         AL(k,i)*W(itemL(k,i),P)
        enddo
        W(i,Q)= VAL
      enddo
      enddo

!$omp end parallel
```

Results, Fortran, 48-threads, 128^3 Time for ICCG, with First-Touch

Implementation	sec.
reorder0	1.288
+ ELL	0.674
+ Reduced OMP Parallel	0.581