Effective Methods in Reducing Communication Overheads in Solving PDE Problems on Distributed-Memory Computer Architectures

Chris Ding and Yun (Helen) He
Lawrence Berkeley National Laboratory

Outline

- Introduction
  - Traditional Method
  - Ghost Cell Expansion (GCE) Method
  - GCE Algorithm
  - Diagonal Communication Elimination (DCE) Technique
- Analysis of GCE Method
  - Message Volume
  - Communication Time
  - Memory Usage and Computational Cost
- Performance
  - Test Problem
  - Test Results
- Conclusions

Background

- On a distributed system, each processor holds a problem subdomain, which includes one or several boundary layers (usually called ghost cells).
- Ghost cells contain the most recent values of the neighboring processors. They must be updated at each time step. And it is done so in traditional approach. The message sizes exchanged between processors are usually very small.
- To speedup the communication, one idea is to combine messages for different time steps and exchange the bigger messages less frequently to reduce the communication latency.

Traditional Field Update

L = 1

ghost | active | ghost

t: O | O O O O | O

t+1: | x x x x |

L = 2

ghost | active | ghost

t: O O | O O O O | O O

t+1: | x x x x |

Where L is the number of layers required depends on the order of accuracy of numerical discretization

Ghost Cell Expansion (GCE) Method

```
              L = 1 , e= 3
       ghost     |  active   |    ghost
t:    O O O O | O O O O | O O O O
t+1:    x  x  x |  x  x  x  x | x  x  x
t+2:       x  x |  x  x  x  x | x  x
t+3:          x |  x  x  x  x | x
t+4:            |  x  x  x  x |
```

```
                      L = 2 , e= 3
          ghost       |  active   |    ghost
t:    O O O O O | O O O O | O O O O O
t+1:   S  x  x  x |  x  x  x  x | x  x  x  S
t+2:      S  x  x |  x  x  x  x | x  x  S
t+3:         S  x |  x  x  x  x | x  S
t+4:              |  x  x  x  x |
```

where e is the expansion level

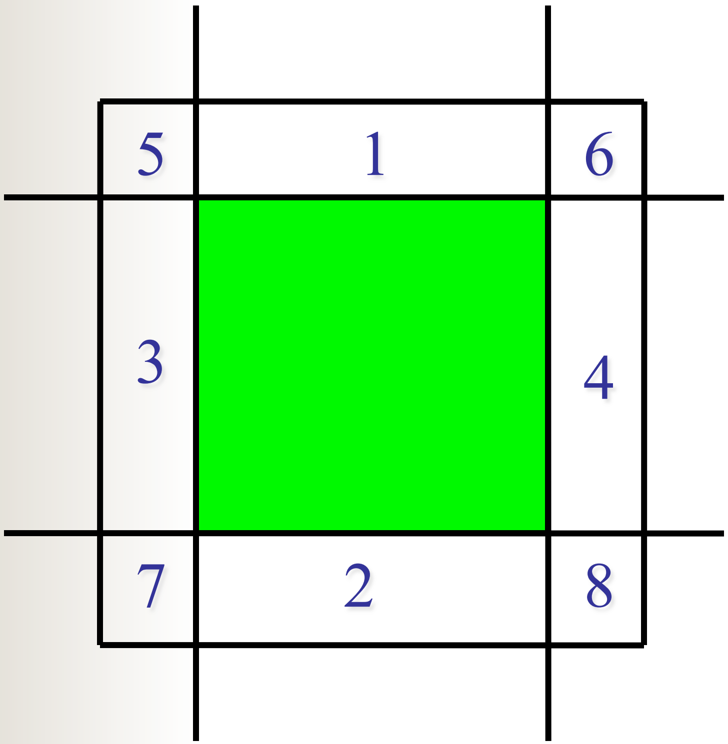
Algorithm for GCE Method

```
do istep = 1, total_steps
    j = mod (istep-1,e+1)
    if (j == 0) update ghost_cells
    y_start = 1 - e + j
    y_end   = ny + e - j
    x_start = 1 - e + j
    x_end   = nx + e – j
    if subdomain touches real boundaries
        set y_start, y_end to 1 or ny
        set x_start, x_end to 1 or nx
    endif
    do iy = y_start, y_end
    do ix = x_start, x_end
        update field (ix, iy)
    enddo
    enddo
enddo
```

Diagonal Communication Elimination (DCE) Technique

- A regular grid in 2D decomposition has 4 immediate neighbors (left and right), and 4 second nearest neighbors (diagonal).
- A regular grid in 3D decomposition has 6 immediate neighbors (horizontal and vertical), 12 second nearest neighbors (planar corner), and 8 third nearest neighbors (cubic corner).

Diagonal Communication Elimination (DCE) Technique *(cont'd)*



Send blocks 5, 7 and 3 right;
Send blocks 6, 8 and 4 left;
Processor owns block 1 also has blocks 5 and 6;
Processors owns block 2 also has blocks 7 and 8;
Send blocks 5, 6 and 1 down;
Send blocks 7, 8 and 2 up;

Active domain updated correctly!

Message Volume

Traditional:

$$V^{old} = 2L(Nx + Ny + 2L)$$

GCE Method:

$$V^{new}(e) = (2L + 2e)(Nx + Ny + 2L + 2e)/(e + 1)$$

Ratio:

$$\frac{V^{new}}{V^{old}} \cong \frac{L + e}{L(e + 1)}$$

$L = 2$, $e = 4$ ➔ ratio = 3/5

Communication Time

Traditional:

$$T^{old} = 2L(Nx + Ny + 2L)8/B + 4T_L$$

GCE Method:

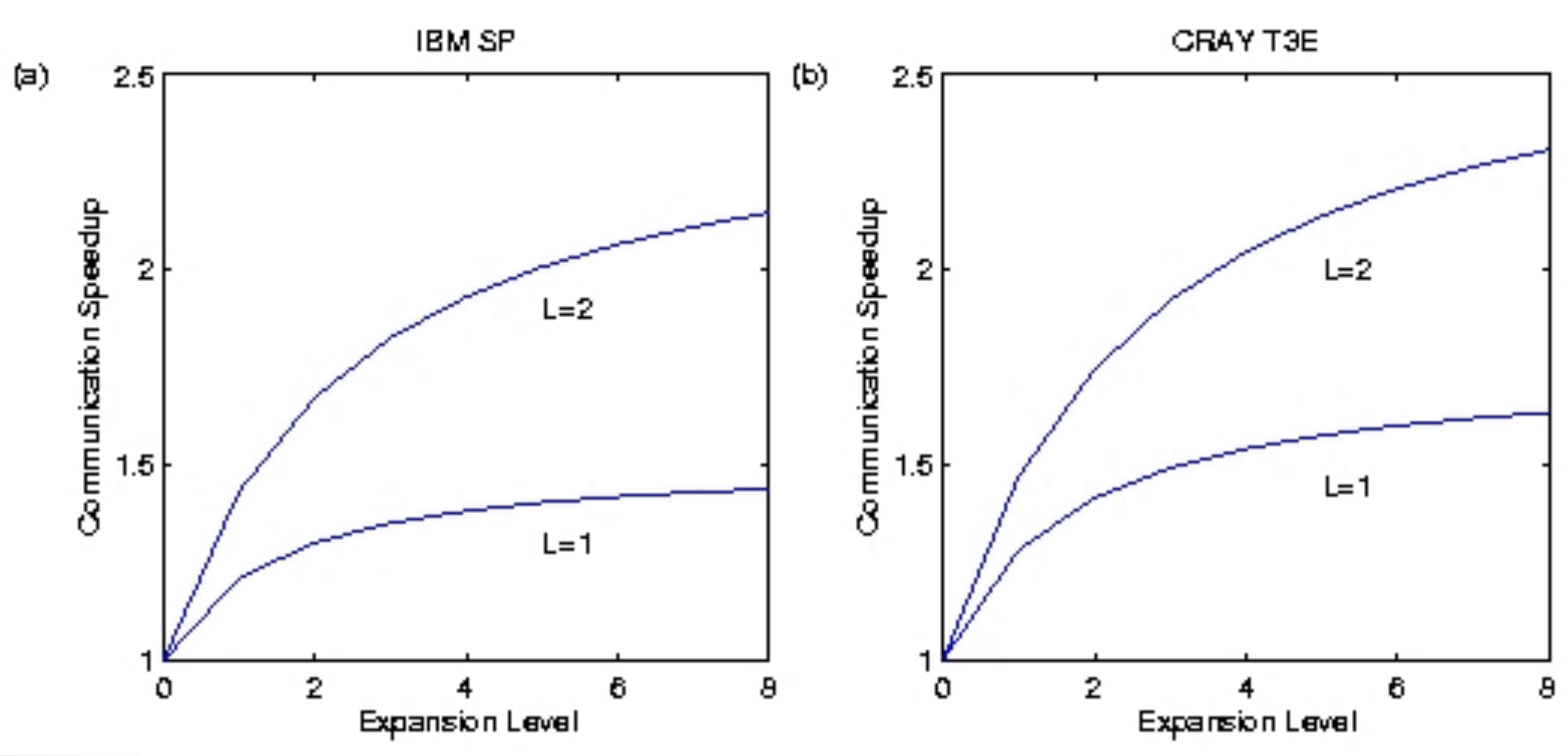
$$T^{new}(e) = [(2L + 2e)(Nx + Ny + 2L + 2e)8/B + 4T_L)/(e+1)$$

L=2, e=8, Nx = Ny = 800:

IBM SP: B=133 MB/sec, T_L=26 μsec → ratio = 2.15

Cray T3E: B=300 MB/sec, T_L=17 μsec → ratio = 2.31

Theoretical Speedup



Memory Usage and Computational Cost

Decomposition	$\Delta M/M$	$\Delta C/C$
1D	$2e/Nx$	$e/2Nx$
2D	$2e/Nx+2e/Ny$	$e/2Nx+e/2Ny$
3D	$2e/Nx+2e/Ny+2e/Nz$	$e/2Nx+e/2Ny+e/2Nz$

$e = 4,\ Nx = Ny = 800 \rightarrow \Delta M/M = 2\%,\ \Delta C/C = 0.5\%$

Test Problem

- 2D Laplacian Equation with Dirichlet boundary conditions:

$$\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} = 0$$

- With 2nd-order accuracy, using 5-point stencils:

$$u(x,y) = \frac{1}{4}[u(x-1,y) + u(x+1,y) + u(x,y-1) + u(x,y+1)]$$

- With 4th-order accuracy, using 9-point stencils:

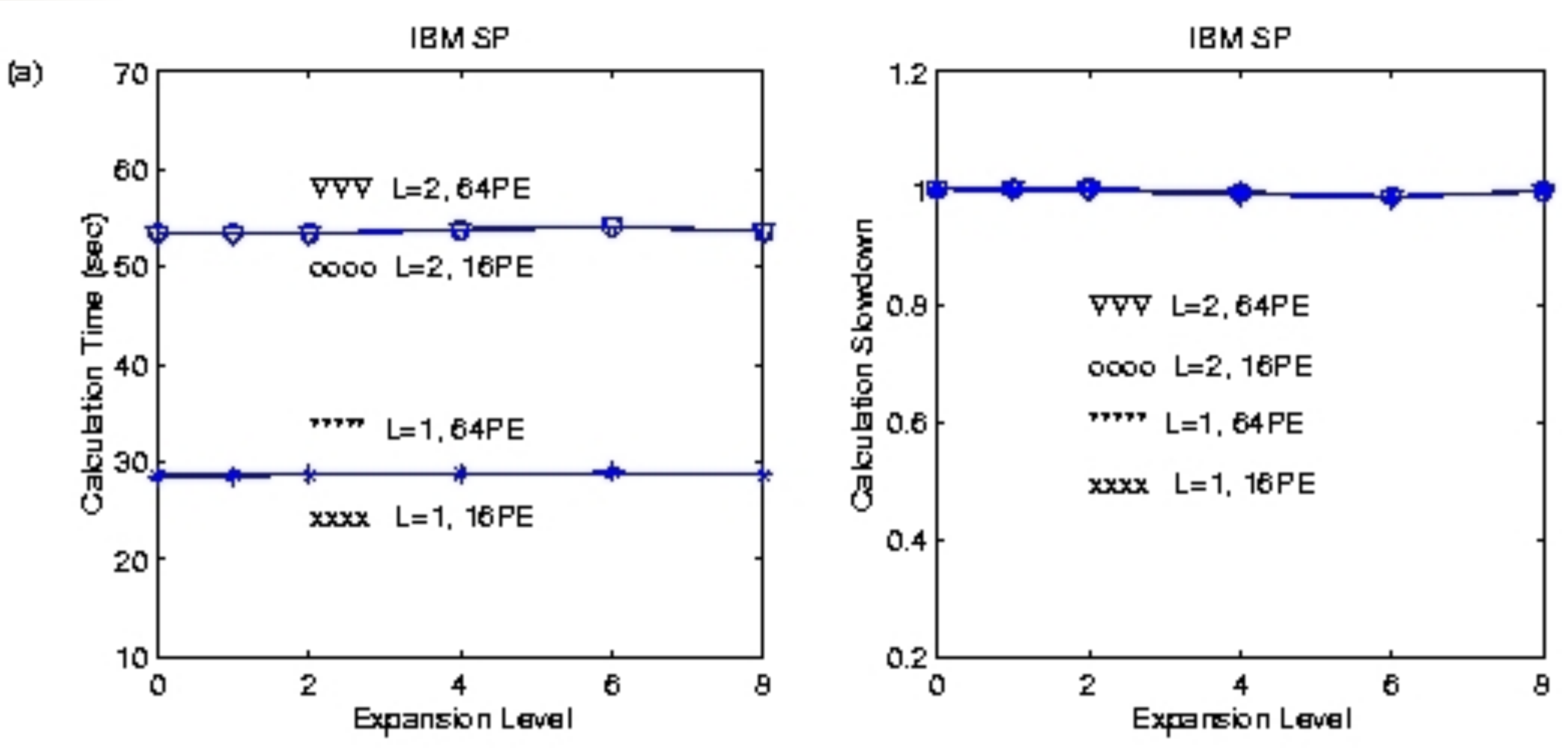
$$u(x,y) = \frac{1}{60}[16u(x-1,y) + 16u(x+1,y) + 16u(x,y-1) + 16u(x,y+1) - u(x-2,y) - u(x+2,y) - u(x,y+2) - u(x,y-2)]$$

Performance Analysis

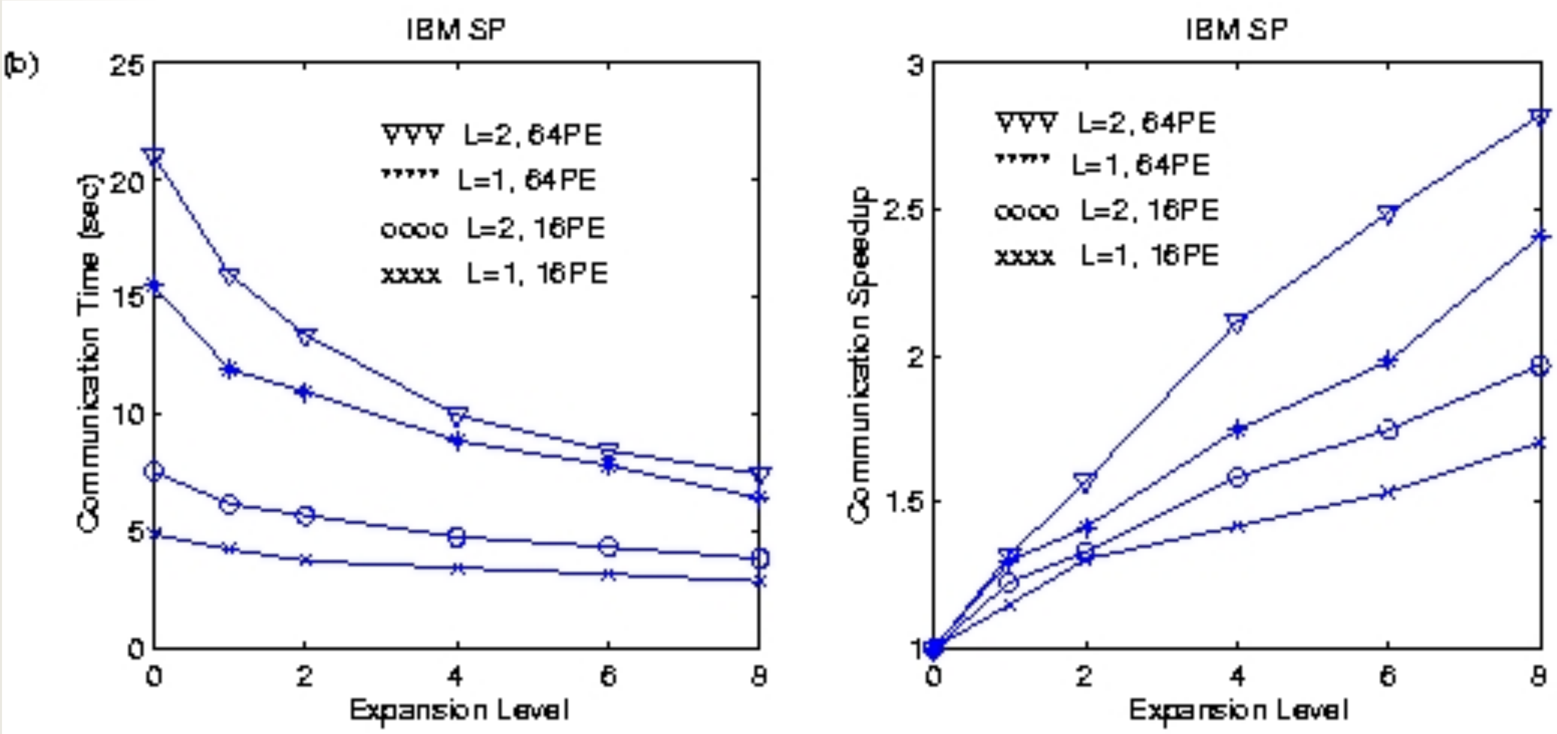
- Test 1: Global size 3200x3200, P=16, L=1
- Test 2: Global size 3200x3200, P=16, L=2
- Test 3: Global size 6400x6400, P=64, L=1
- Test 4: Global size 6400x6400, P=64, L=2

Local domain size is always 800x800

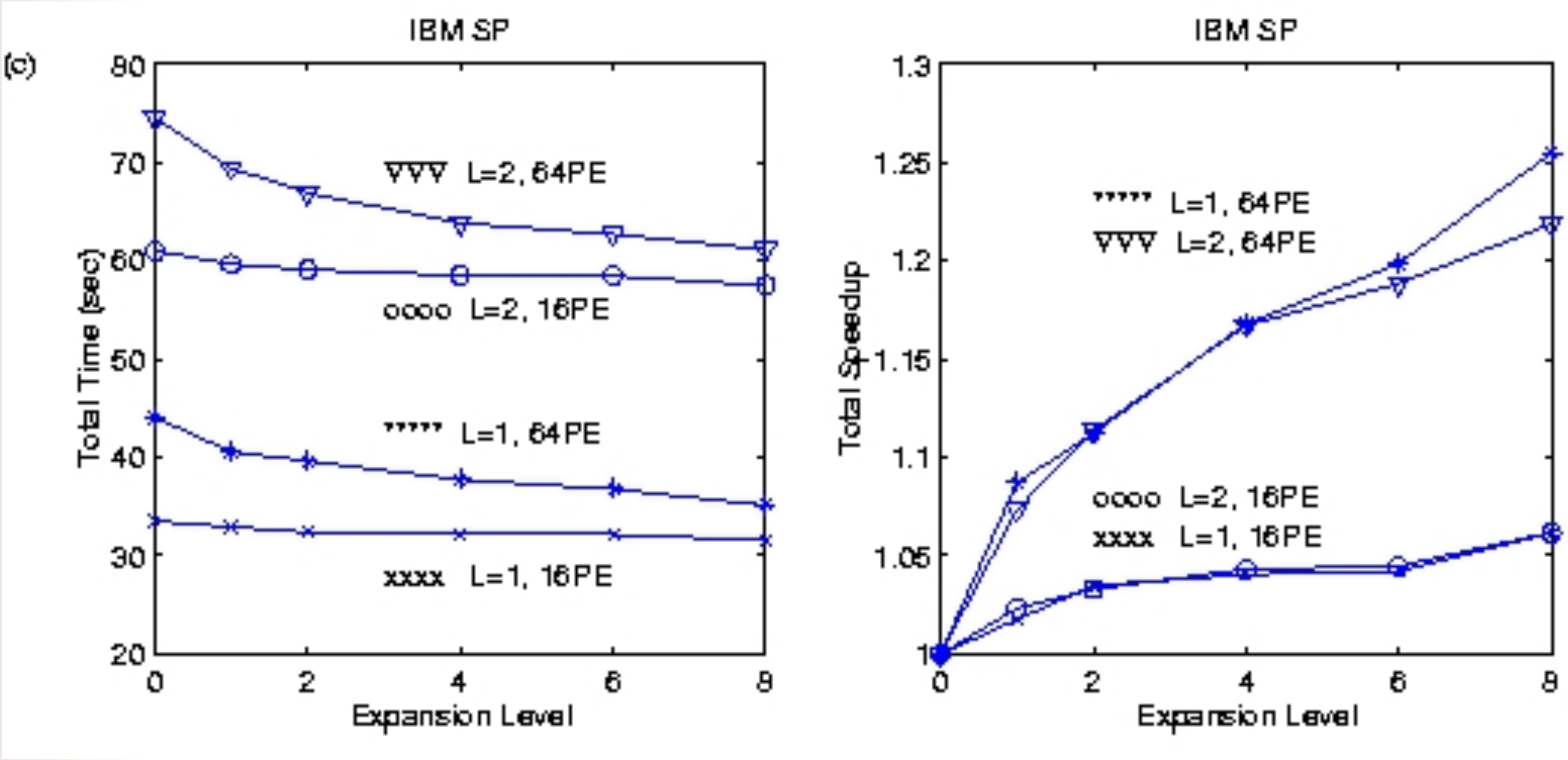
Performance on IBM SP



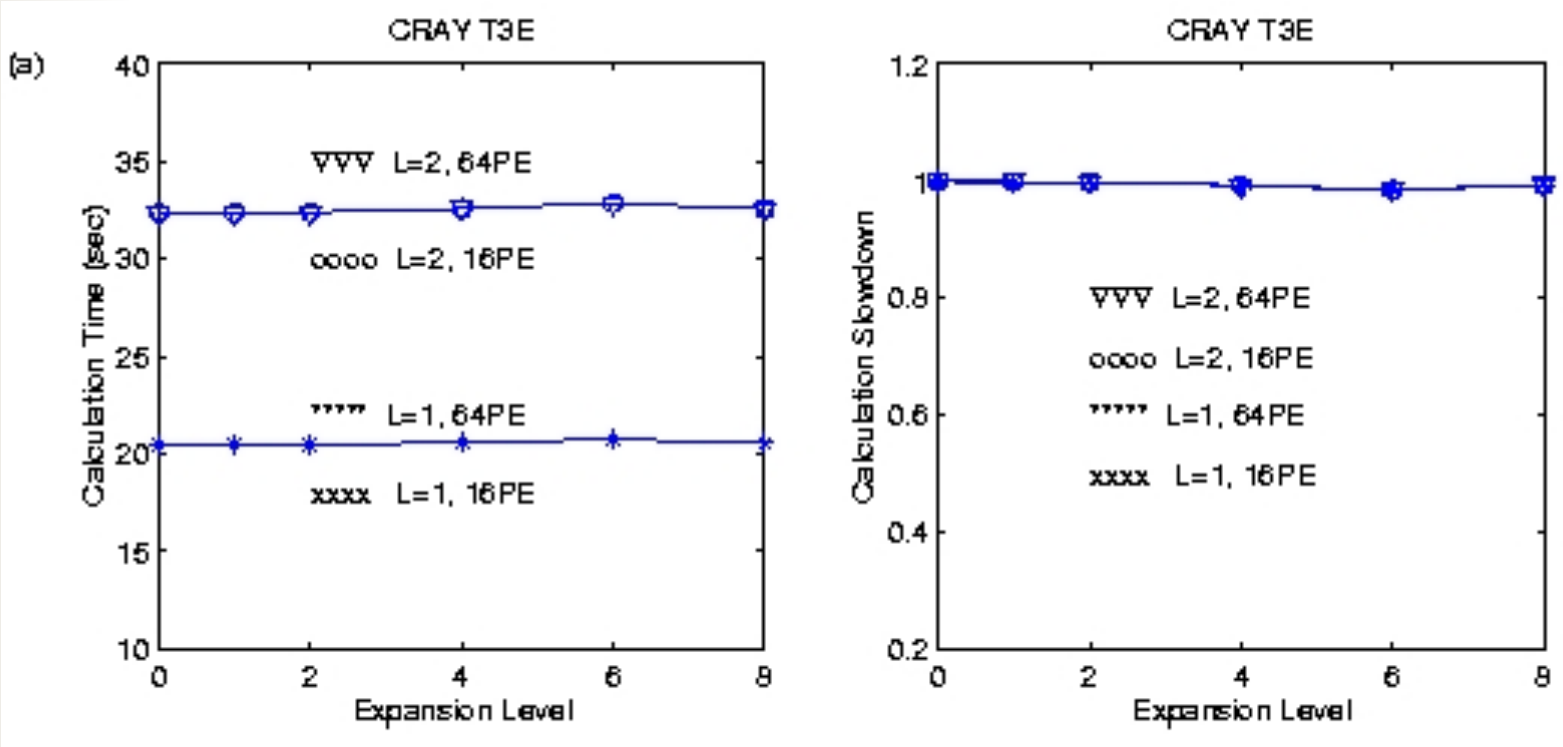
Performance on IBM SP (cont' d)



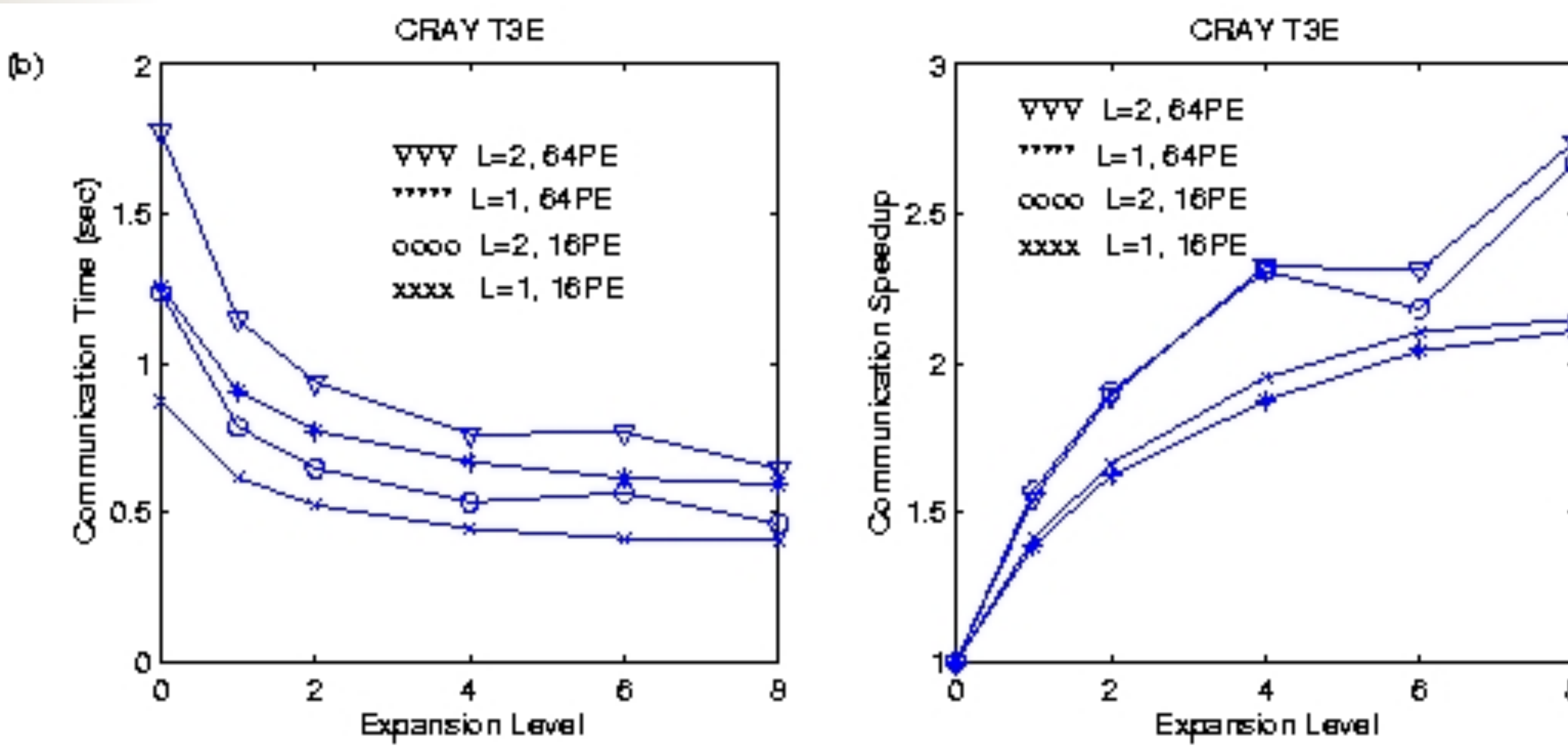
Performance on IBM SP (cont' d)



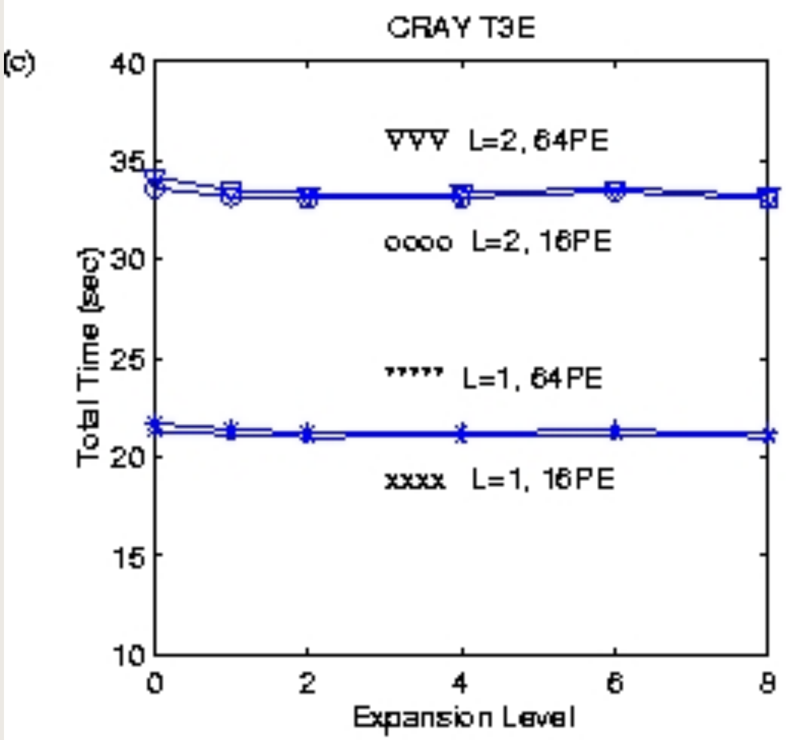
Performance on Cray T3E

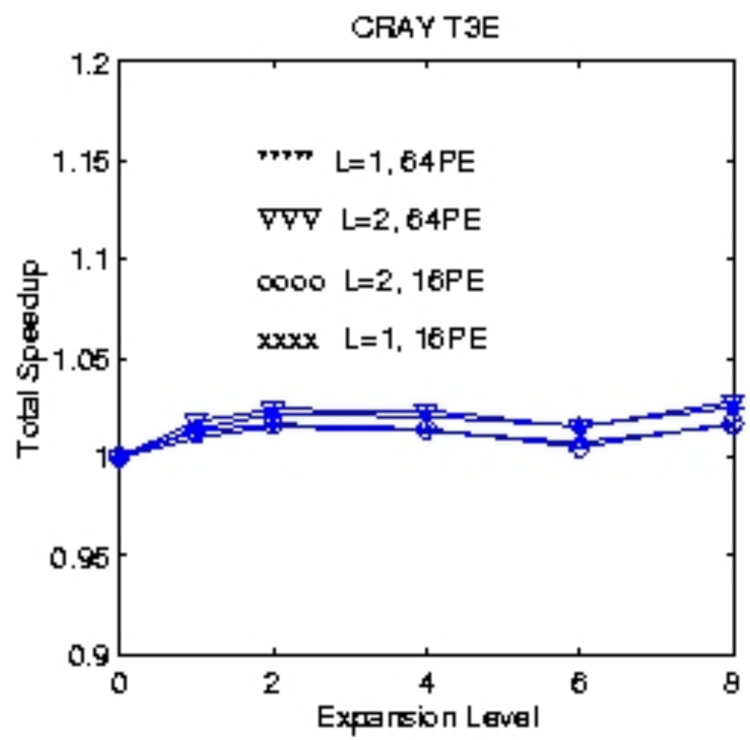


Performance on Cray T3E (cont’d)



Performance on Cray T3E (cont’d)





Conclusions

- With the new ghost cell expansion method, the frequency to update processor subdomain boundaries is reduced from every time step to once per e+1 time steps.
- Both the total message volume and total number of messages are reduced, thus the total communication time. The overhead of memory usage and computational cost are minimal.
- Diagonal Communication Elimination technique reduces the total number of messages exchanged between processors from 8 to 4 in 2D domain decomposition and 26 to 6 in 3D domain decomposition.
- The systematic experiments on IBM SP and Cray T3E both have communication speedup up to 170%.