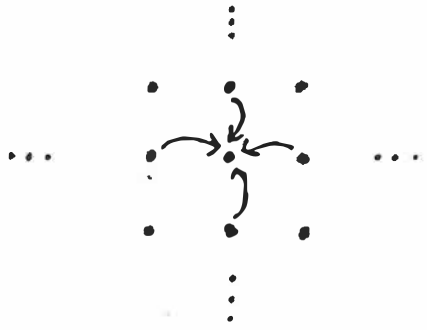
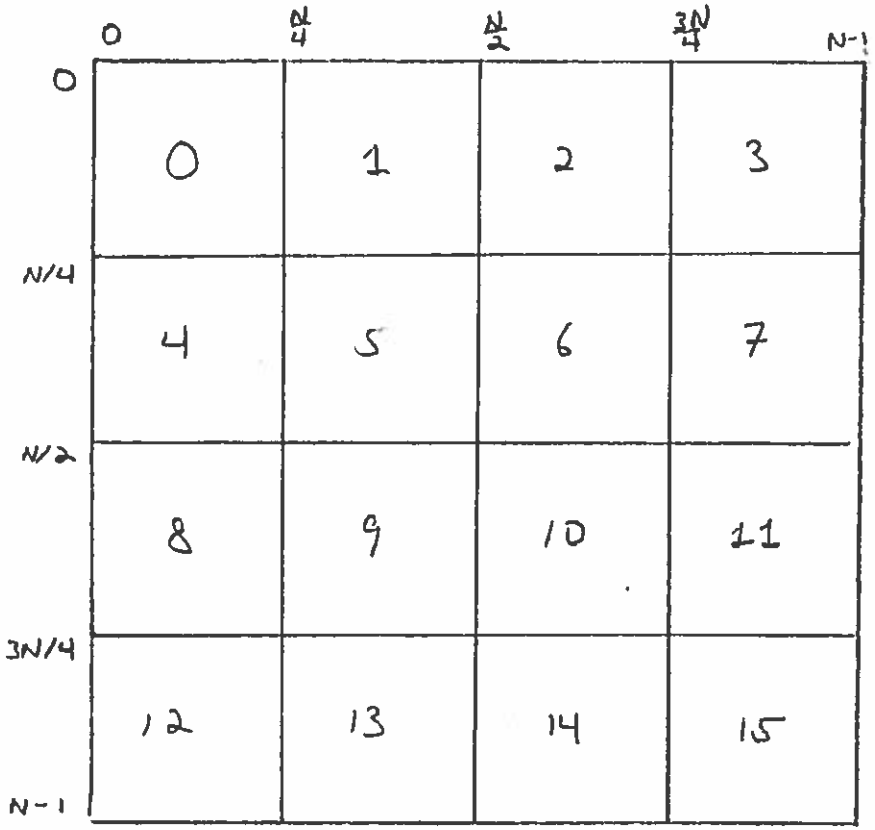
Parallel Application Example: Jacobi Relaxation



```
double A[N][N], B[N][N], **temp;

main() {
   int i, j, k;

   initialize(A);
   for(i=0; i< n.iterations; i++) {

        for(j= 1; j < N-1; j ++)
              for (k = 1;  k < N-1; k++)
                   B[j][k] = (A[j-1][k] + A[j+1][k] + A[j][k-1] + A[j][k+1])
                                  / 4.0;

        temp = A;
        A = B;
        B = temp;
   }
}
```

Jacobi Solver

Ex: N x N Jacobi on 16 processors



Jacobi is an ideal parallel application.

- Block partition
- Communicate border values
- Synchronize after computing all blocks

Parallel Shared Memory (3)
Jacobi Example

```
double A[N][N], B[N][N], **temp;

void
work(int my_pid)
{
  int i, j, k;
  int chunk_size, kstart, kend, jstart, jend, rootP;

  rootP = (int)sqrt(P);
  chunk_size = N / rootP;

  kstart = (my_pid % rootP) * chunk_size;
  kend = (my_piad % rootP + 1) * chunk_size;

  jstart = (my_pid / rootP) * chunk_size;
  jend = (my_pid / rootP + 1) * chunk_size;

  if (kstart == 0) kstart = 1;
  if (kend == N) kend = N-1;
  if (jstart == 0) jstart = 1;
  if (jend == N) jend = N-1;

  for (i = 0; i < n_iterations; i++) {
    for (j = jstart; j < jend; j++) {
      for (k = kstart; k < kend; k++) {
        b[j][k] = (A[j-1][k] + A[j+1][k] + A[j][k-1] + A[j][k+1]) /
          4.0;
      }
    }

    temp = A;
    A = B;
    B = temp;

    barrier(NUM_PROCS);
  }
}

int
main()
{
  int i;

  initialize(A);
  for (i = 1; i < NUM_PROCS; i++)
    create_thread(i, work, i);
  work(0);
}
```