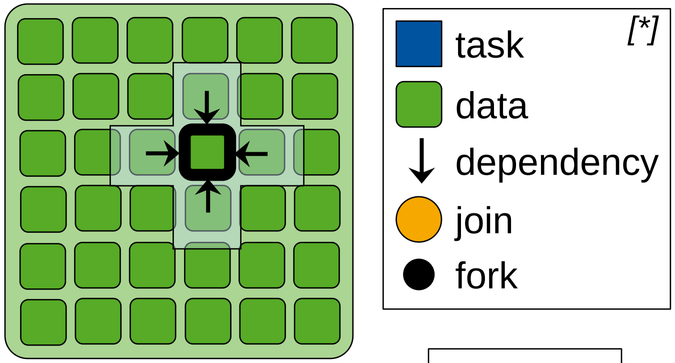
Stencil

- Access of neighboring input elements with fixed offsets
- Ability for data reuse & cache optimization



OpenACC

```
#pragma acc parallel
#pragma acc loop tile(64,4) gang vector
for(i=1; i<n-1; i++) {
  for(j=1; j<m-1; j++) {
  #pragma acc cache(a[i-1:3][j-1:3])
  anew[i][j] = (a[i-1][j] + a[i+1][j] + \
         a[i][j-1] + a[i][j+1]) * 0.25;
} }
```

- **tile: strip-mining**
  - → 1st no. = block size of most inner loop
- **cache: data caching**
  - → Just hint, compiler can ignore
  - → (Performance-wise) needed especially for software-managed mem (GPUs)

OpenMP

```
#pragma omp target
#pragma omp teams distribute collapse(2)
for(i=1; i<n-1; i+=4) {
  for(j=1; j<m-1; j+=64) {
  #pragma omp parallel for collapse(2)
    for(k=i; k<min(n-1,i+4); k++) {
      for(l=j; l<min(m-1,j+64); l++) {
        anew[k][l] = (a[k-1][l] + ^\
               a[k+1][l] + a[k][l-1] + \
               a[k][l+1]) * 0.25;
} } } }
```

- **Tiling must be expressed explicitly**
  - → More development effort than w/ `tiling`
  - → Performance expected similar to `tiling`
- **No caching hints possible**
  - → Maybe performance loss on GPUs

[*] Similar to: McCool, M., Reinders, J., Robison, A.: Structured Parallel Programming: Patterns for Efficient Computation. Morgan Kaufmann, first edn. (2012)