Parallel Program Design

Dr. Tassadaq Hussain

RIPHAH
INTERNATIONAL
UNIVERSITY

Microsoft Research
Centre

- Applications Types
- PCAM Methodology
- Decomposition Patterns
- Program Structure

Applications

Compute Intensive
Data Intensive
Complex and Irregular

C and C++ Applications

http://people.sc.fsu.edu/~jburkardt/cpp_src/cpp_src.html

http://people.sc.fsu.edu/~jburkardt/c_src/c_src.html

PCAM Methodology

Partitioning

Communication

Agglomeration

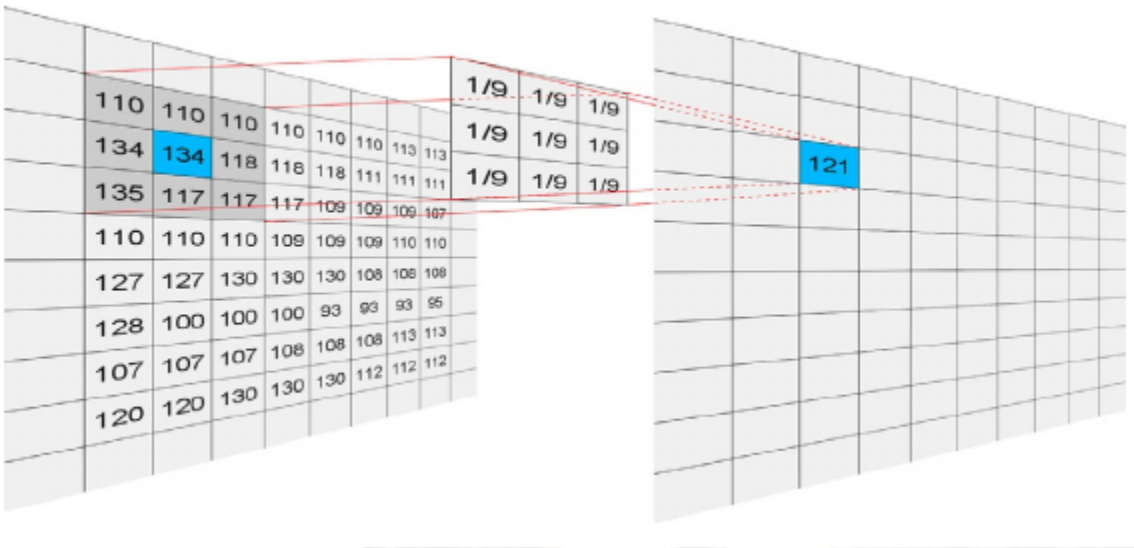
Mapping

Application Understanding

Metamathematical Representation

$$g(x,y) = \sum_{i=-n2}^{n2} \sum_{j=-n2}^{n2} k(n2+i, n2+j) f(x-i, y-j)$$

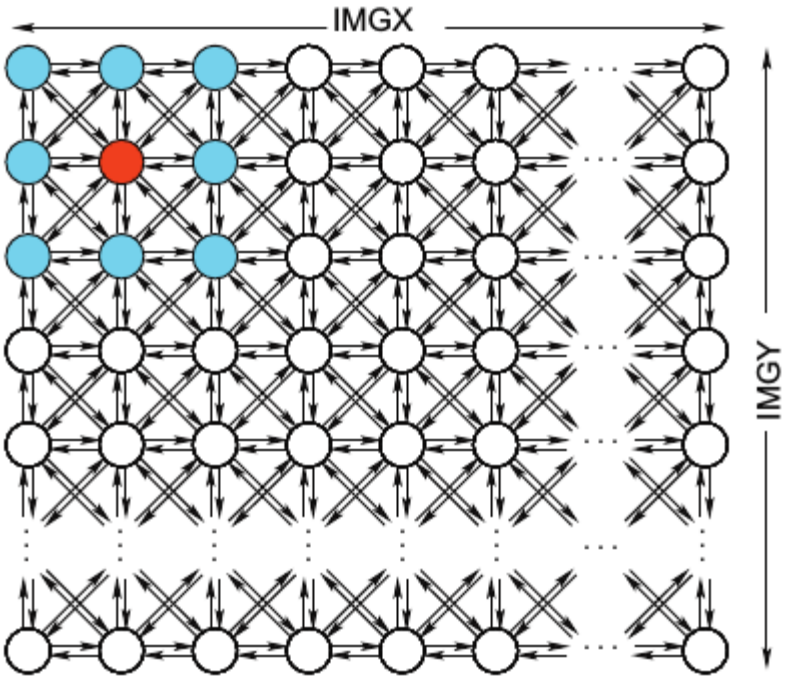
Working Operation



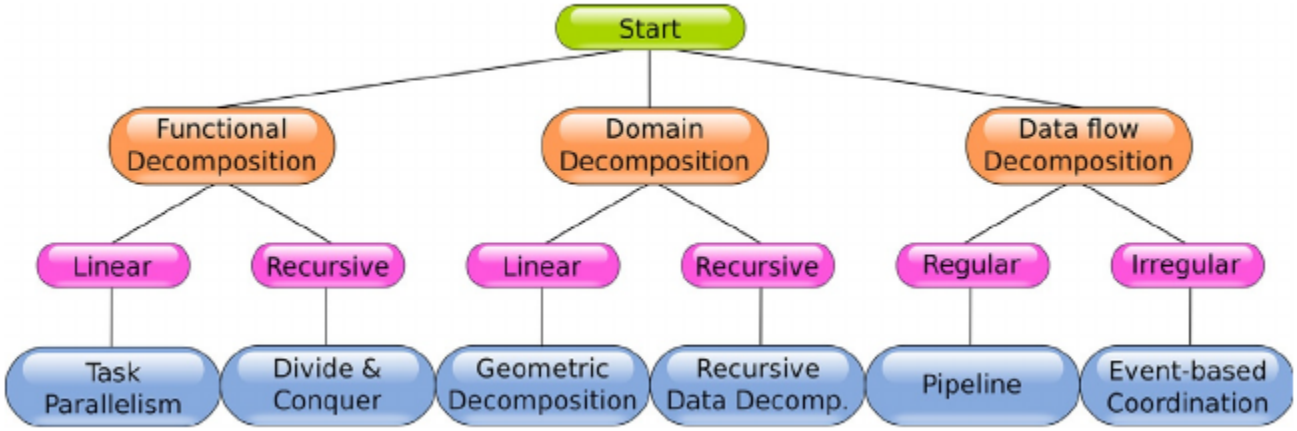
Pseudocode

```
1   int img[IMGY+2][IMGX+2];
2   int filt[IMGY][IMGX];
3   int n2 = n/2;
4   for(int x=1;x <= IMGX; x++) {
5      for(int y=1; y <= IMGY ; y++)  {
6         int newV=0;
7         for(int i= -n2; i<= n2; i++)
8            for(int j= -n2; j<= n2; j++)
9               newV += img[ y - j][ x - i ] * k[n2 + j][n2 + i];
10        filt[y-1][x-1] = newV;
11        }
12     }
```

Decomposing Application



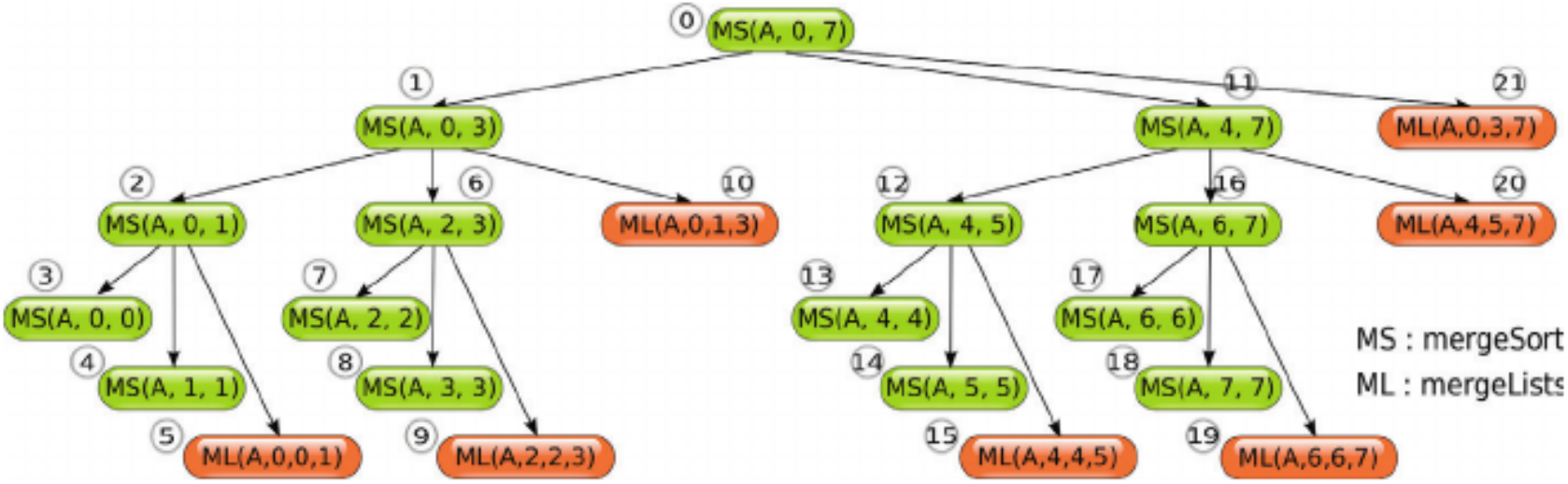
Decomposition

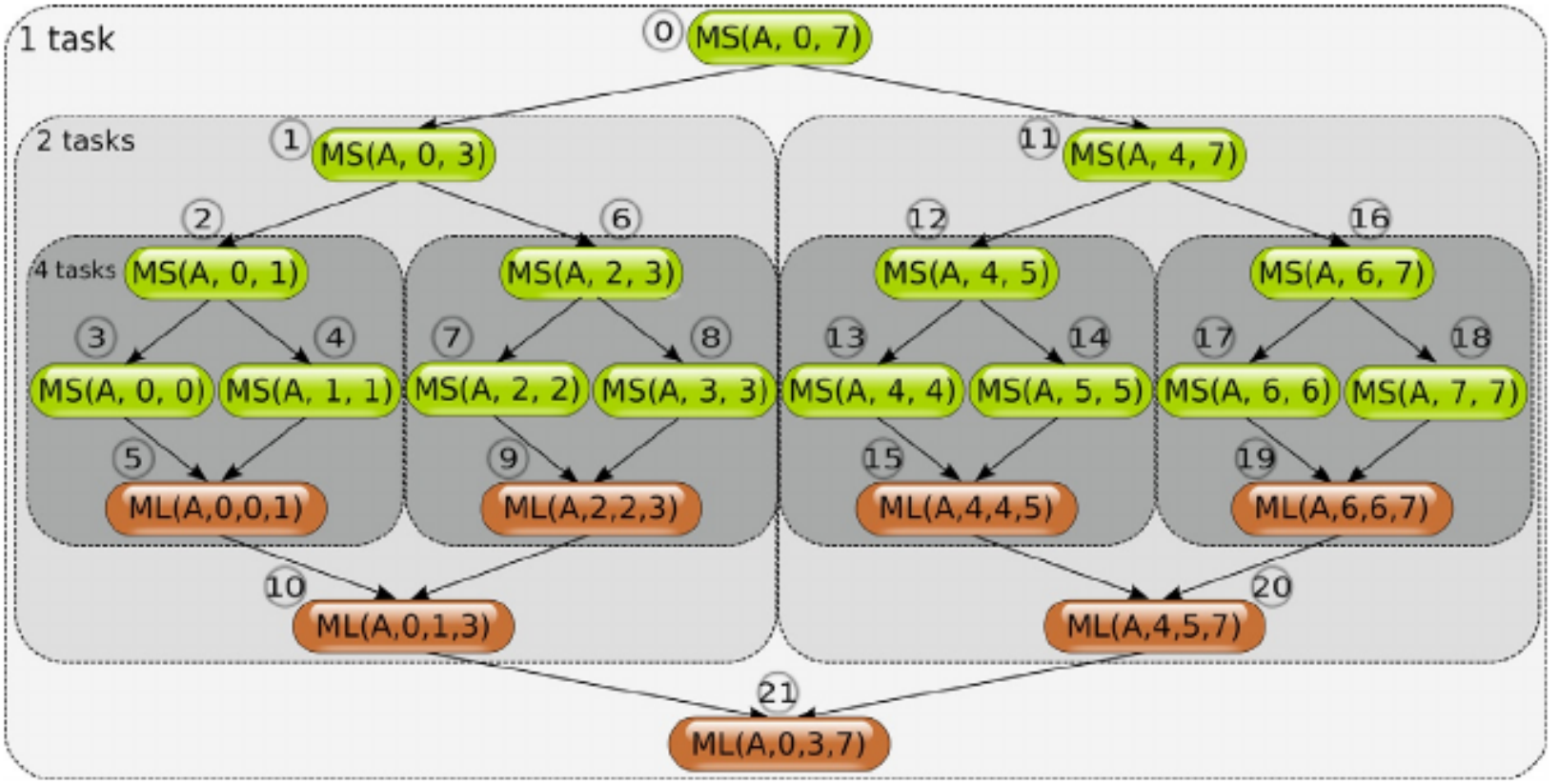


Types of Decomposition

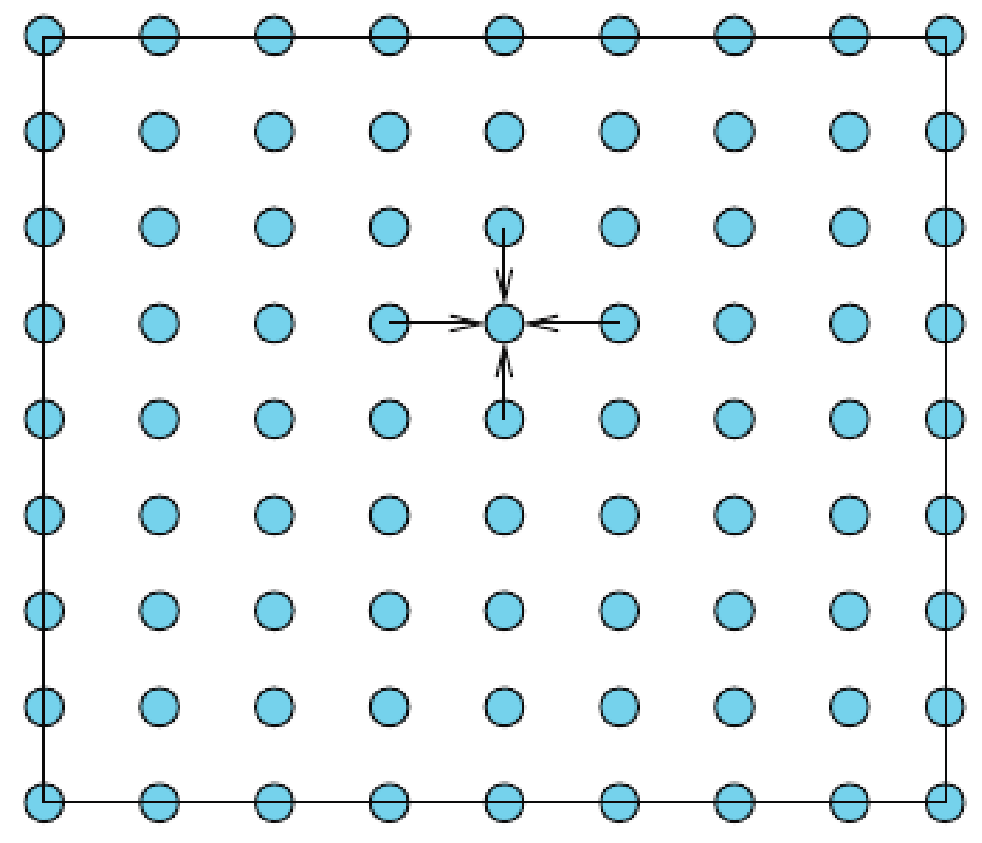
- Functional Decomposition
  - Task Parallelism
  - Divide & Conquer
- Domain Decomposition
  - Geometric
  - Recursive Data
- Data Flow Decomposition
  - Pipelining
  - Event Based

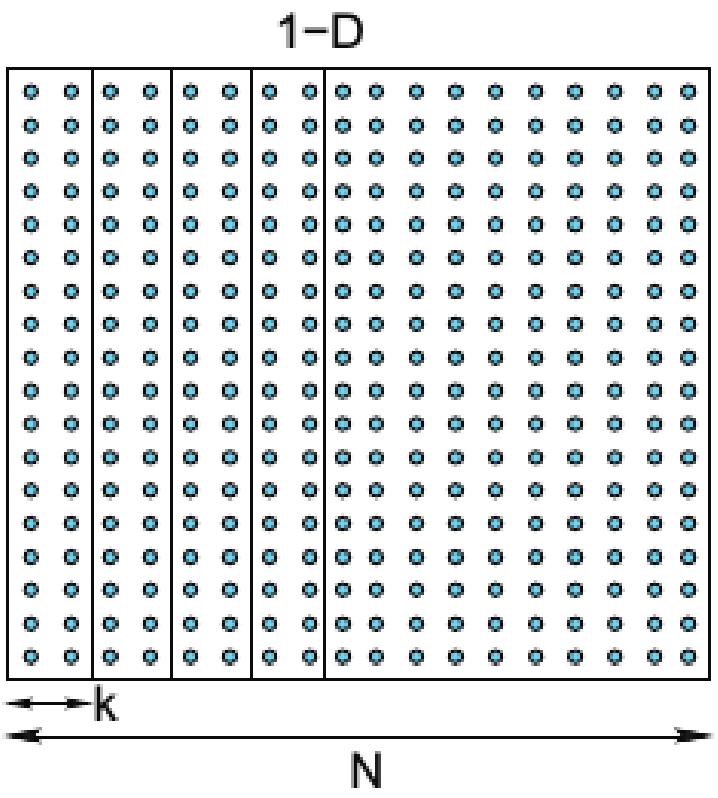
Divide & Conquer

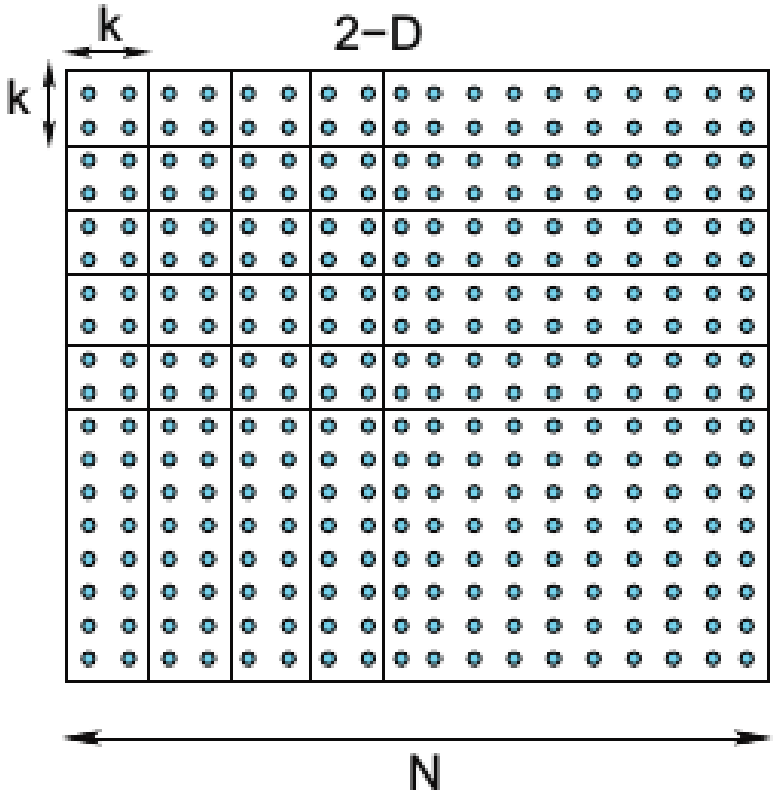



1 task
0
MS(A, 0, 7)
2 tasks
1
MS(A, 0, 3)
11
MS(A, 4, 7)
2
6
12
16
4 tasks
MS(A, 0, 1)
MS(A, 2, 3)
MS(A, 4, 5)
MS(A, 6, 7)
3
4
7
8
13
14
17
18
MS(A, 0, 0)
MS(A, 1, 1)
MS(A, 2, 2)
MS(A, 3, 3)
MS(A, 4, 4)
MS(A, 5, 5)
MS(A, 6, 6)
MS(A, 7, 7)
5
9
15
19
ML(A,0,0,1)
ML(A,2,2,3)
ML(A,4,4,5)
ML(A,6,6,7)
10
20
ML(A,0,1,3)
ML(A,4,5,7)
21
ML(A,0,3,7)

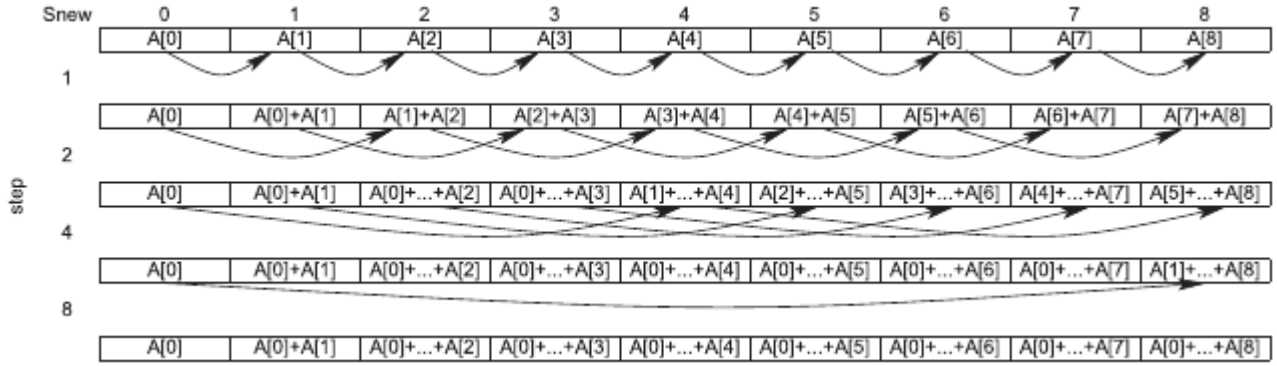
Geometry Decomposition




1−D
k
N

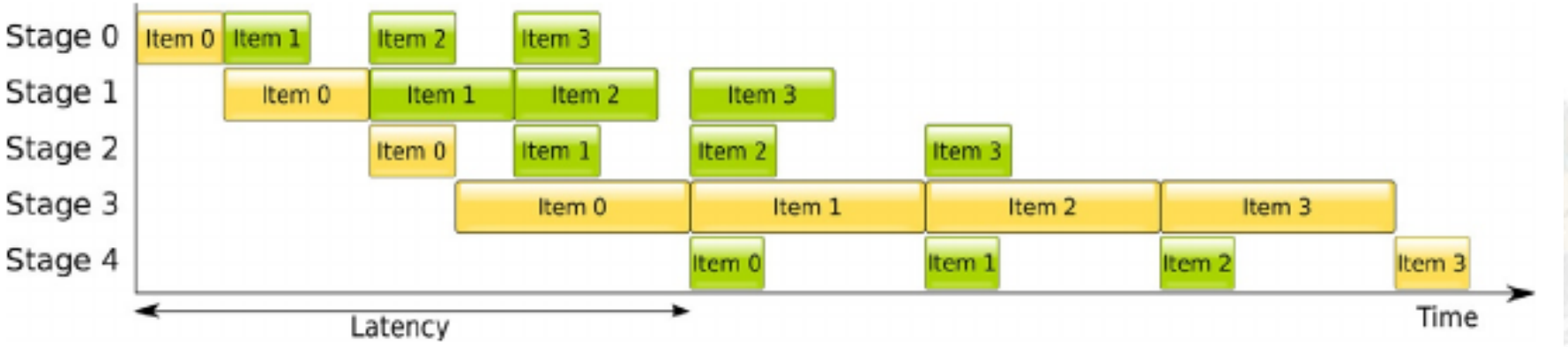

k
k
2−D
N

Recursive Domain Decomposition



Pipeline Decomposition

Depends on Processor Architecture



Event Based

An event is a time-stamped message that can represent a status change in the state of a module, a trigger to change the state, a request to perform an action, a response to a previously generated request, or the like.

Program Structure

Globally Parallel, Locally Sequential (GPLS):

GPLS means that the application is able to perform multiple tasks concurrently, with each task running sequentially.

Patterns that fall in to this category include:

- Single program, multiple data
- Multiple program, multiple data
- Master-worker
- Map-reduce

Globally Sequential, Locally Parallel (GSLP):

GSLP means that the application executes as a sequential program, with individual parts of it running

in parallel when requested.

Patterns that fall in to this category include:

- Fork/join
- Loop parallelism