Unal Center of
Education Research
and Development
www.ucerd.com

Applications & Problems

Dr. Tassadaq Hussain
Assistant Professor Riphah International University

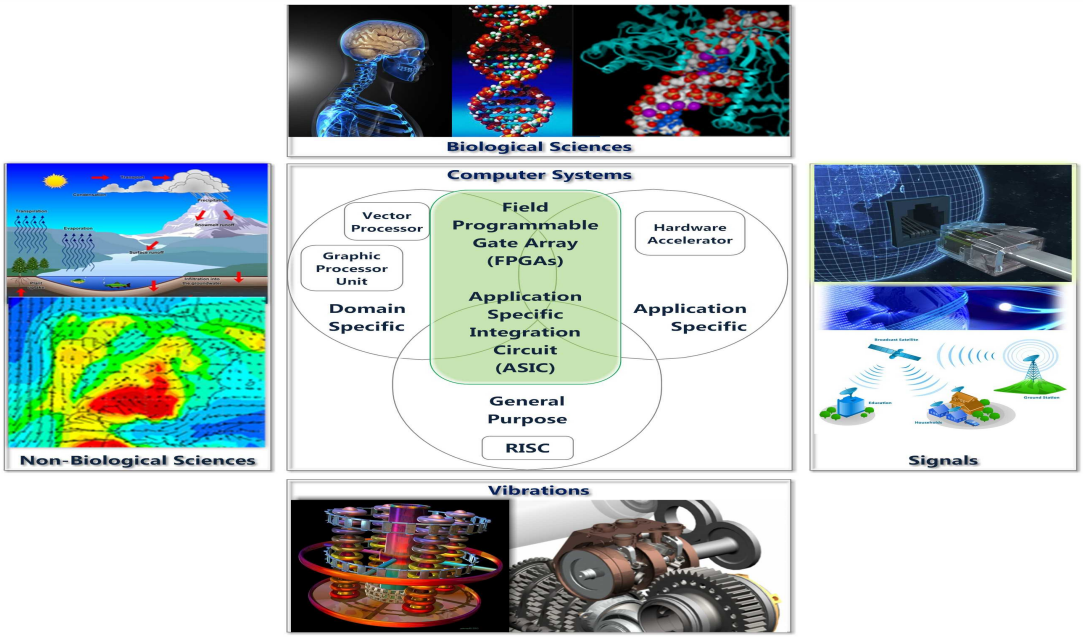
Collaborations:
Barcelona Supercomputing Center Barcelona, Spain
UCERD Pvt Ltd Islamabad

BSC~Microsoft Research Centre

- Applications/Problems
  - Categories
  - Data Access Patterns
  - Parallel Execution Methods

Applications

- Life Sciences
  - Biomedical Applications
  - Imaging Applications
  - Communication
  - Defense
- Earth Sciences
  - Interferometric Sensors
  - Oil Search

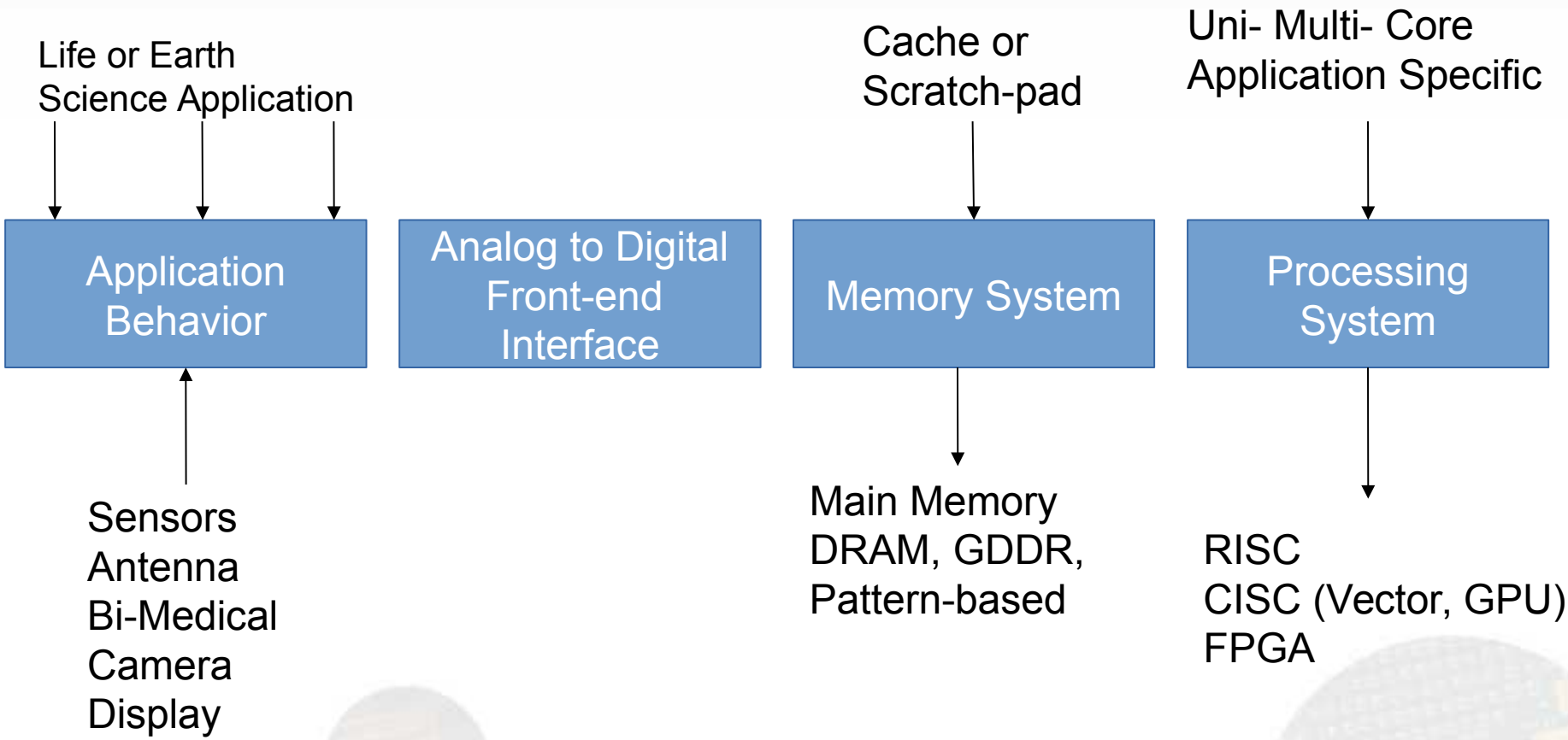


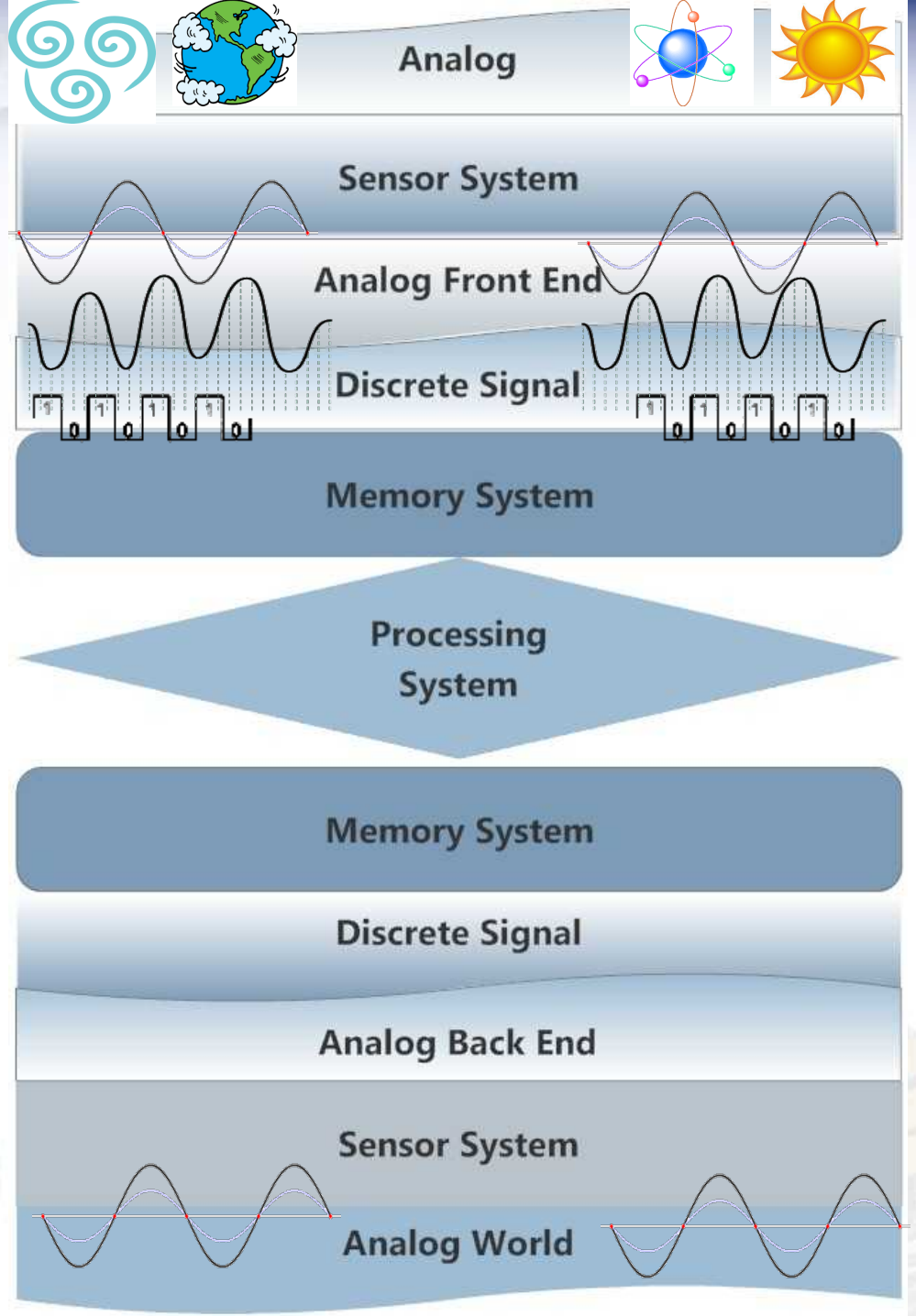
Computer Application

- Access Pattern
  - Front-end Interface
- Storage
- Processing

Application Behavior	Analog to Digital Front-end Interface	Memory System	Processing System

Understand an Application




Analog
Sensor System
Analog Front End
Discrete Signal
Memory System
Processing System
Memory System
Discrete Signal
Analog Back End
Sensor System
Analog World

High Performance Computing (HPC) Application

Complex and Irregular Transfer

Compute and Data Intensive

Basic types of memory access patterns

- Regular access
  - Fixed stride
  - Predictable
  - Parallel
- Irregular access
  - Variable strides
  - Known
    - » Predictable at compile-time
  - Unknown
    - » Independent
    - » Dependent

```
data[1024];
  for(int x=y;x<100;x=x++)
     { read=data[x];
      compute(read);
     }
```

Regular access pattern

```
data[1024];
  for(int x=0;x<5;x=x++)
    {  read=data[factorial(x)];
       compute(read);  }
```

Irregular known access pattern

```
Data[1024];
addr=runtime_input();
  for(int x=0;x<5;x++)
     { read=data[factorial(x)+addr];
       compute(read); }
```

Irregular unknown independent access pattern

```
data[100];
  for(int x=0;x<100;x=x++)
     {  read=data[read+x];
        compute(read);
     }
```

Irregular unknown dependent access pattern

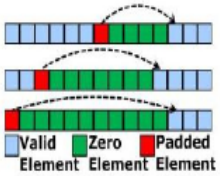
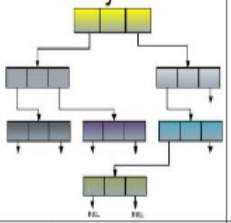
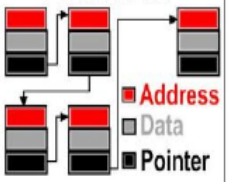
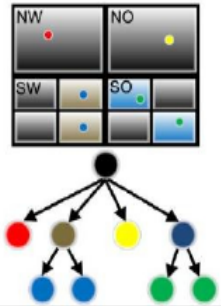
Basic types of memory access patterns

- Regular access
  - Fixed stride
  - Predictable
  - Parallel
- Irregular access
  - Variable strides
  - Known
    - » Predictable at compile-time
  - Unknown
    - » Independent
    - » Dependent

Kernel	Description	Access Pattern
Rad_Con	Radian Converter converts degree into radian	Load/Store
Thresh	Thresholding is an application of image segmentation, which takes streaming 8-bit pixel data and generates binary output.	
FIR	Finite Impulse Response calculates the weighted sum of the current and past inputs.	Streaming
FFT	Fast Fourier Transform is used for transferring a time-domain signal into corresponding frequency-domain signal.	1D Block
Mat_Mul	Matrix Multiplication takes pair of tiled data and produce Output tile. Output= Row[Vector] × Column[Vector] X=Y×Z	Column & Vector Access
Smith_W	Smith-Waterman determines the optimal local alignments between nucleotide or protein sequences.	Diagonal Access
Lapl	Laplacian kernel applies discrete convolution filter that can approximate the second order derivatives.	2D Tiled
3D-Sten	3D-Stencil algorithm averages nearest neighbor points (size 8x9x8) in 3D.	3D Stencil

Basic types of memory access patterns

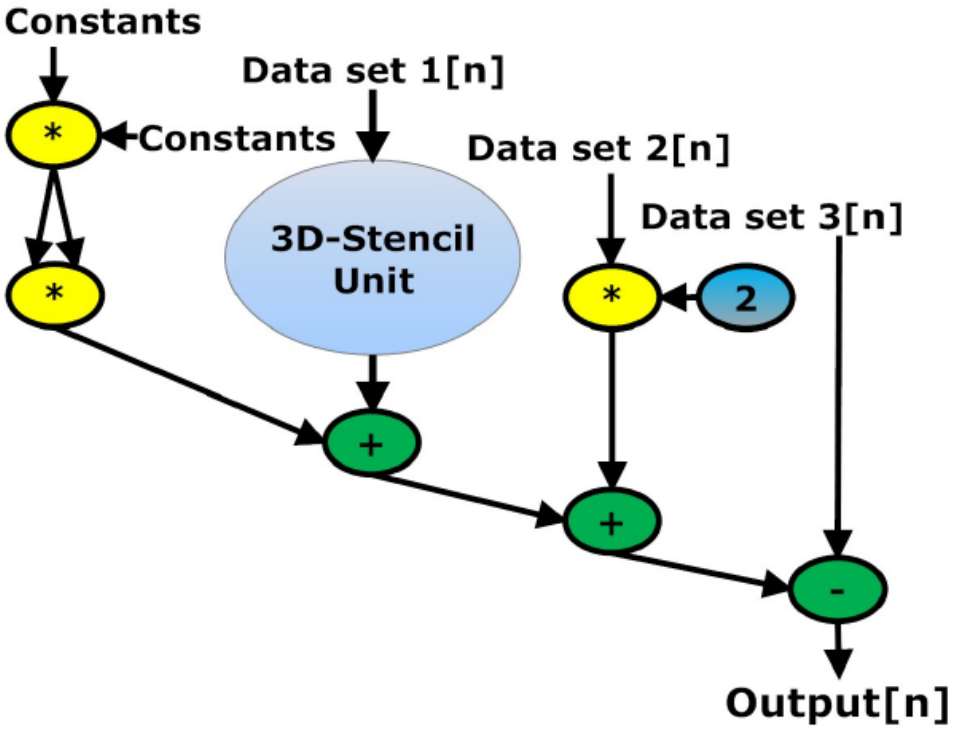
- Regular access
  - Fixed stride
  - Predictable
  - Parallel
- Irregular access
  - Variable strides
  - Known
    - » Predictable at compile-time
  - Unknown
    - » Independent
    - » Dependent

Kernel	Description	Access Pattern	Regular %	Irregular % known	Irregular % unknown
CRG	A compression algorithm, hides zero in a descriptor block	Pointer 		28	72
Huffman	Huffman is an entropy coding technique. Allocate codes to symbols, using frequency of occurrence for each symbol.	Binary Tree 		25	75
In_Rem	A Linked List Buffer	Linked List 	5	55	40
N-Body	The 3D-Hermite algorithm used to compute movement of bodies using the newtonian gravitational force.	Tree 	20	40	40

Compute and Data Intensive

Arithmetic Intensity

Control and Data Flow



Executing Application on Parallel Machines

Partitioning: Divide the computation to be performed and the data operated on by the computation into small tasks. The focus here should be on identifying tasks that can be executed in parallel.

Communication: Determine what communication needs to be carried out among the tasks identified in the previous step.

Agglomeration or aggregation: Combine tasks and communications identified in the first step into larger tasks. For example, if task A must be executed before task B can be executed, it may make sense to aggregate them into a single composite task.

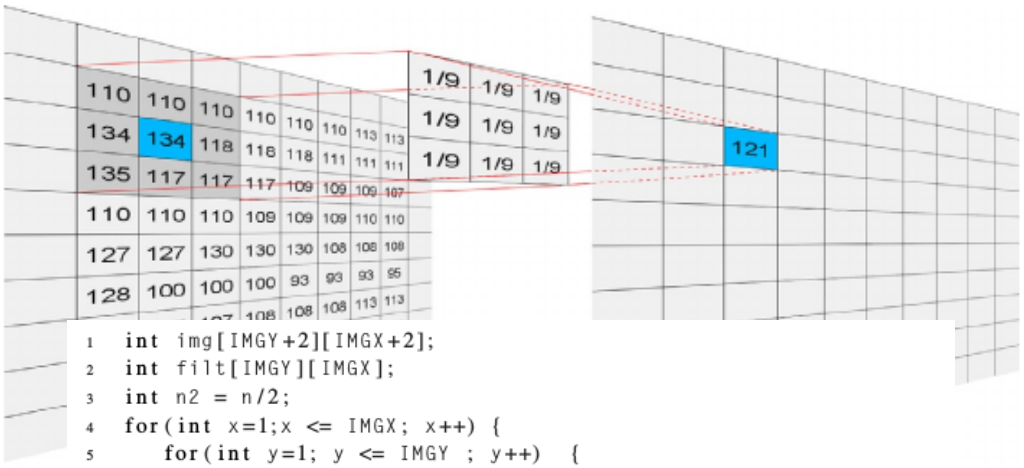
Mapping: Assign the composite tasks identified in the previous step to processes/threads. This should be done so that communication is minimized, and each process/thread gets roughly the same amount of work.

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

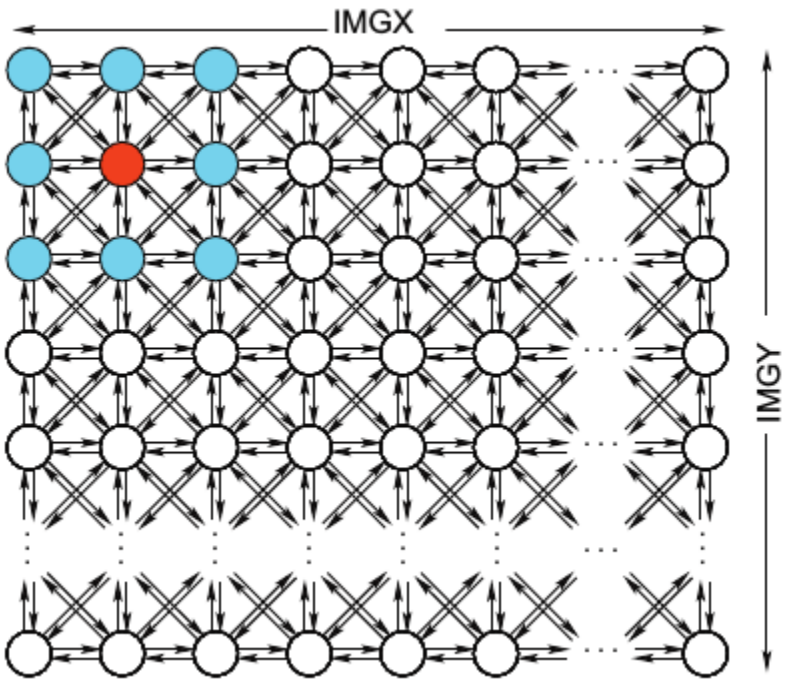



```
int img[IMGY+2][IMGX+2];
int filt[IMGY][IMGX];
int n2 = n/2;
for(int x=1;x <= IMGX; x++) {
   for(int y=1; y <= IMGY ; y++)  {
      int newV=0;
      for(int i= -n2; i<= n2; i++)
         for(int j= -n2; j<= n2; j++)
            newV += img[ y - j][ x - i ] * k[n2 + j][n2 + i];
      filt[y-1][x-1] = newV;
      }
   }
```

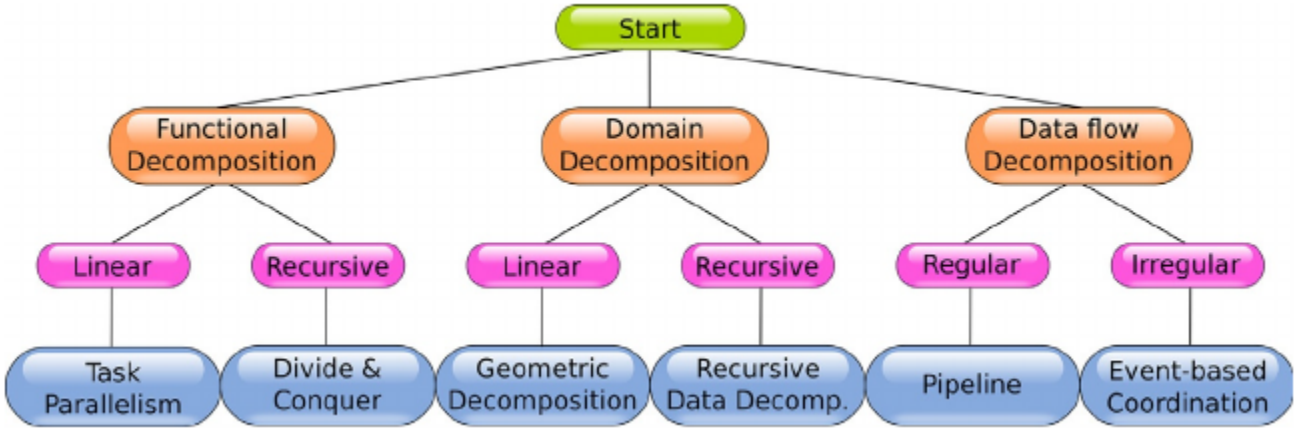
Computational Intensity

Floating Point Operations / Second ❑ Data Bytes / Second

Decomposing Application



Decomposition



Types of Decomposition

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

Computer Program Structure

Globally Parallel, Locally Sequential (GPLS):

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

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.