# Assignment: Distributed Memory: representation and algorithm

## 1 Heat Equation - 1D

One dimensional heat equation is the simplest example of a stencil computation. It computes iteratively the following equation for a stencil of size $N$.

$$Heat^k[0] = \frac{2Heat^{k-1}[0] + Heat^{k-1}[1]}{3}$$

$$Heat^k[N-1] = \frac{2Heat^{k-1}[N-1] + Heat^{k-1}[N-2]}{3}$$

$$Heat^k[i] = \frac{Heat^{k-1}[i-1] + Heat^{k-1}[i] + Heat^{k-1}[i+1]}{3}, \forall 0 < i < N-1$$

(Assume network topology is a clique.)

### 1.1 Round Robin Decomposition

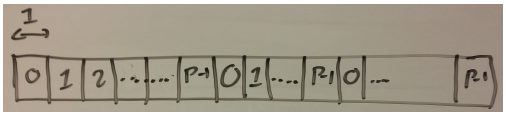



Figure 1: Round Robin Decomposition

**Question:** Write the algorithm that computes heat equation using a Round Robin decomposition.
**Question:** How much communication happen per iteration of the heat equation for a Round Robin decomposition?

### 1.2 Block Decomposition

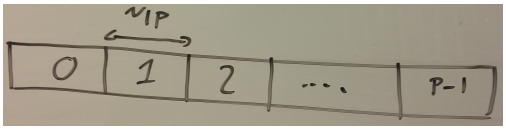



Figure 2: Block Decomposition

**Question:** Write the algorithm that computes heat equation using a Block decomposition.
**Question:** How much communication happen per iteration of the heat equation when using a Block decomposition?

### 1.3 Reflection

**Question:** What data partitioning would you use?

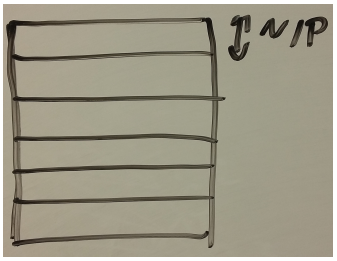
## 2 Dense Matrix Multiplication

Given a matrix $A$ of size $N \times N$ and a vector $x$ of size $N$, the value $y = Ax$ is given by $y[i] = \sum_j A[i][j]x[j]$. Or in other words, to compute $y[i]$ multiply element wise the $i$th row of the matrix by $x$ and sum the values.

(Assume the network topology is a clique.)

Use only blocking Point to Point communication.

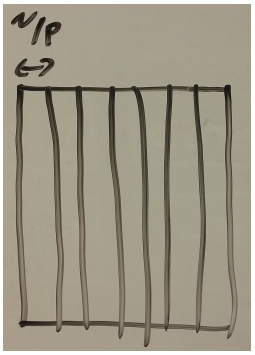
### 2.1 1D partitioning: Horizontal stripes




Horizontal Data Partitioning

**Question:** Write the algorithm that performs $y = Ax; x = y;$ 10 times in a loop if the data is partitioned horizontally.
**Question:** How much memory does each node need if the data is partitioned horizontally?
**Question:** How much communication does the algorithm do per iteration if the data is partitioned horizontally?

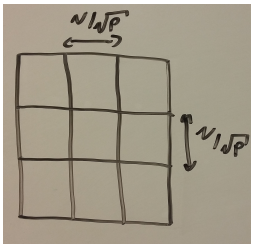
### 2.2 1D partitioning: vertical stripes




Vertical Data Partitioning

**Question:** Write the algorithm that performs $y = Ax; x = y;$ 10 times in a loop if the data is partitioned vertically.
**Question:** How much memory does each node need if the data is partitioned vertically?
**Question:** How much communication does the algorithm do per iteration if the data is partitioned vertically?

### 2.3 2D partitioning: blocks




Block Partitioning

**Question:** Write the algorithm that performs $y = Ax; x = y;$ 10 times in a loop if the data is partitioned in blocks.
**Question:** How much memory does each node need if the data is partitioned in blocks?
**Question:** How much communication does the algorithm do per iteration if the data is partitioned in blocks?

# 3 Reduction

Here are the three most popular reduction algorithms:

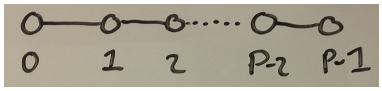
```
reduce-star(p, P, val) {
  if (p == 0) {
    for (i=1; i<P; ++i) {
      recv vald from i;
      val += vald;
    }
  }
  else {
    send val to 0;
  }
}
```

```
reduce-chain(p, P, val) {
  if ( p != P-1) {
    recv vald from p+1;
    val += vald;
  }
  if (p != 0) {
    send val to p-1;
  }
}
```

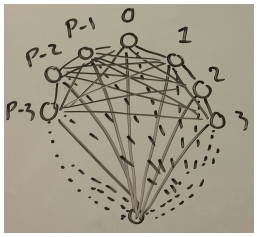
```
//assume P is a power of 2
reduce-tree(p, P, val) {
  fakeP = P;

  while (p < fakeP) {
    if (p >= fakeP/2) {
      send val to p-fakeP/2;
    }else {
      recv valp from p+fakeP/2;
      val += valp;
    }
    fakeP = fakeP / 2;
  }
}
```

Consider the following two network structures:




(a) chain




(b) clique

**Question:** Fill the following table. For each algorithm and each network structure, answer the following questions. Run a small example if you have difficulty seeing how communication happens; but express all answers for the case with P processors.

| Case | How much data on most loaded link | How much data on most loaded node | How long is the longest chain of communication |
|---|---|---|---|
| Reduce-star on chain | | | |
| Reduce-star on clique | | | |
| Reduce-chain on chain | | | |
| Reduce-chain on clique | | | |
| Reduce-tree on chain | | | |
| Reduce-tree on clique | | | |

**Question:** What do you think is the best algorithm for each network structure? (One of the given algorithm or a different one.)