Programmability and Performance of M++ Self-Migrating Threads

October 10th, 2001

Munehiro Fukuda [1,2]

Naoya Suzuki [2]

Luis Miguel Campos [3]

Shinya Kobayashi [4]

1. University of Washington, Bothell
2. University of Tsukuba
3. University of California, Irvine
4. Ehime University

Overview

1. Motivation
2. M++ self-migrating threads
3. Programmability comparison with MPI
4. Performance comparison with MPI
5. Conclusions

Multi-Agent Approach

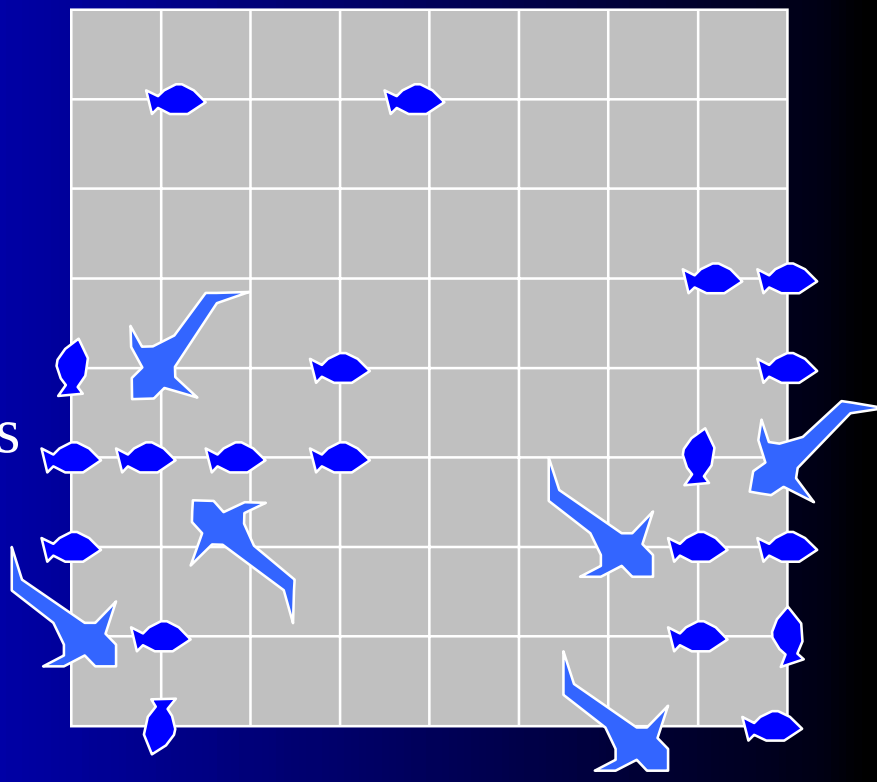
Simulate the interaction among agents

Advantages:

- Overcomes the limitations of mathematical techniques
- Encapsulates simulation models
- Allows open-ended simulation

Disadvantages:

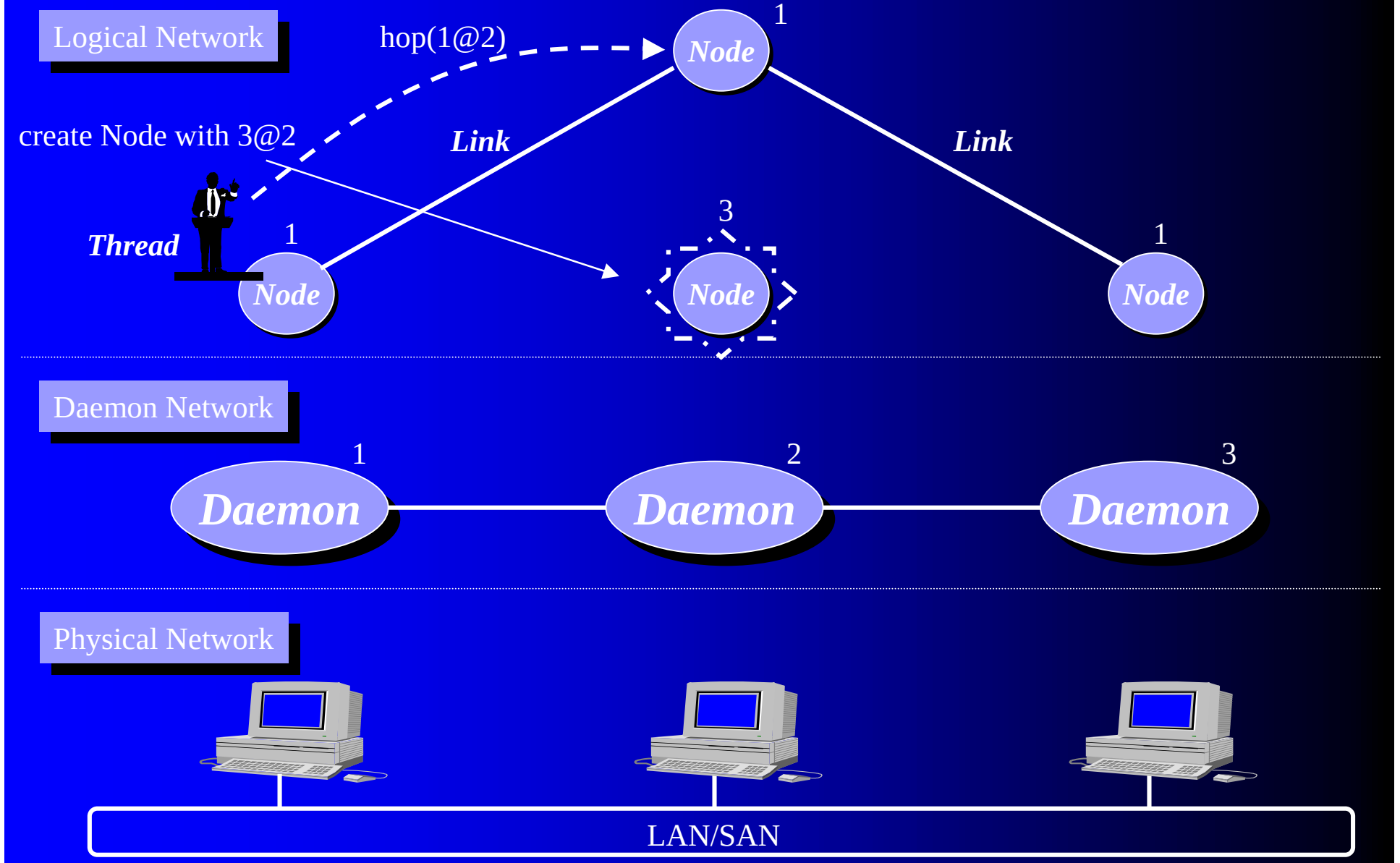
- Performance
- Parallel programming



Research Goal

1. Enhance performance
   - M++ self-migrating threads
   - Zero-copy migration library
2. Support flexible agent programming
   - Logical network construction capability
   - M++ agent-based language

Network Architecture



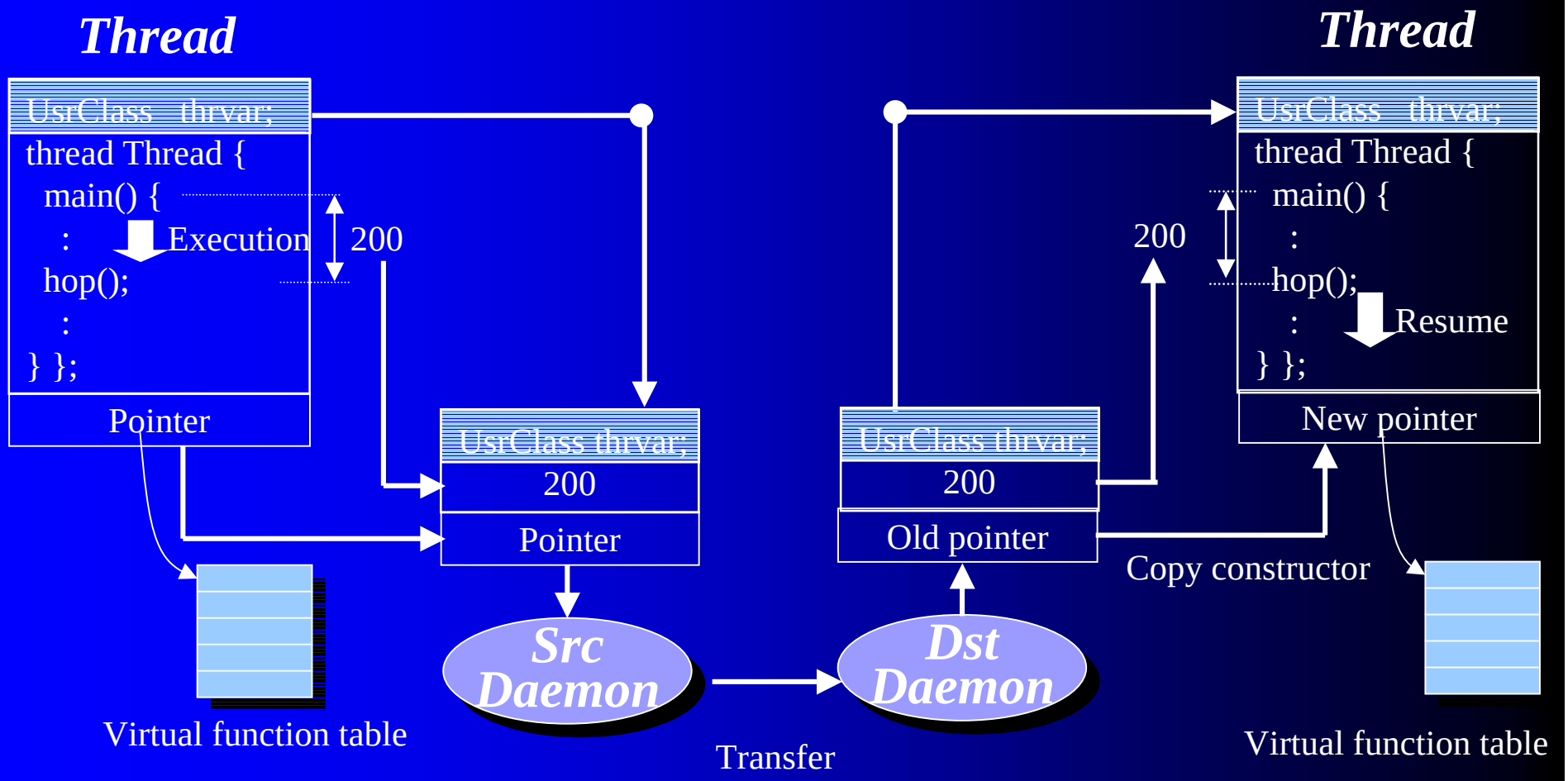
M++ Language

```
class Node {                                  // Defining nodes (and links) as C++ clas
public:
  int var;
};

thread Thread {                               // Defining an M++ thread
private:
  main() {                                    // Executing its own behavior independently
    create node<Node> with 1;                 // Creating a network node, (link, and thread)
    hop( 1 );                                 // Migrating itself (and forking)
    node<Node>.var = 10;                      // Accessing the  current node, (and link)
  }
};
```

Implementation

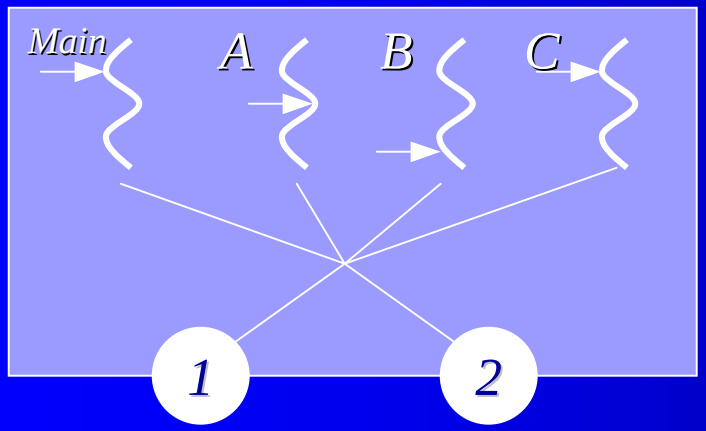
- State Capturing -



Implementation

- Thread Scheduling -

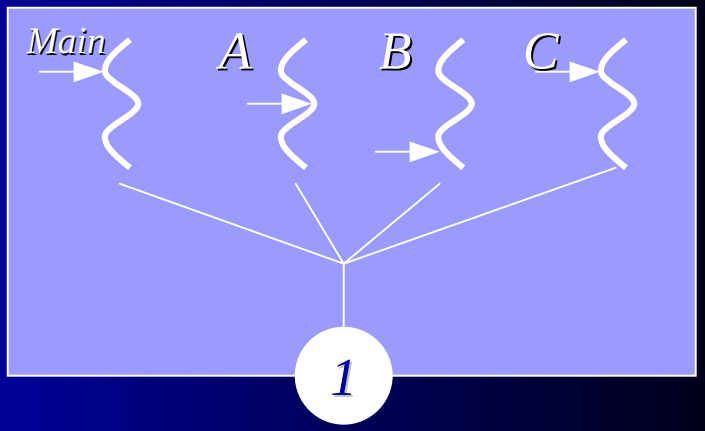
Pthread Library

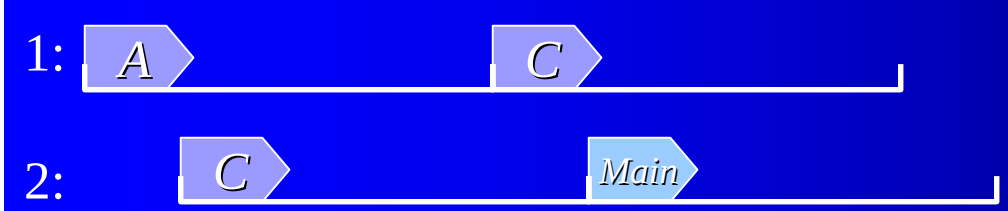


User level

Kernel level

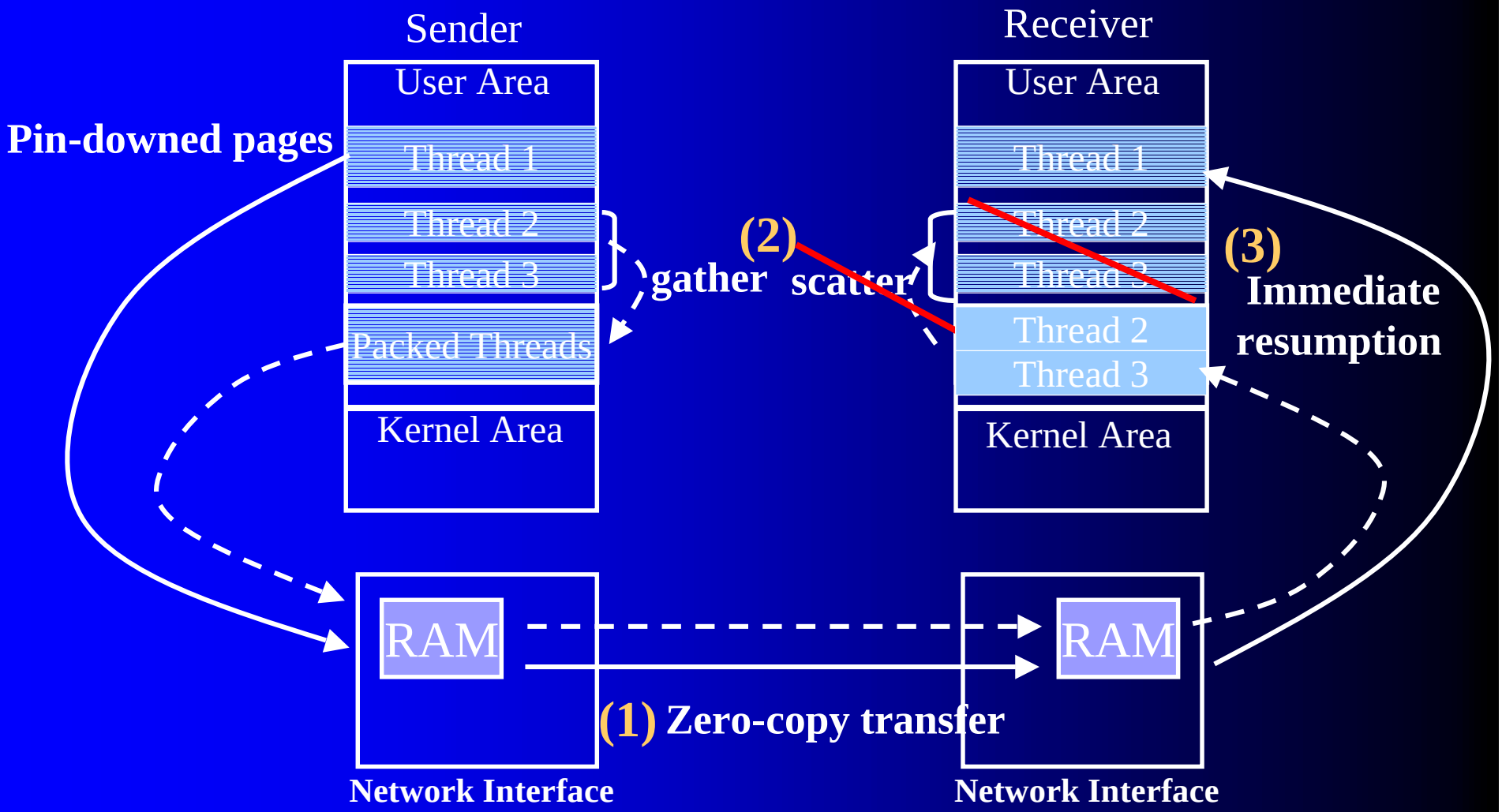
Sthread Library





Implementation

- Communication -



Ant Farm

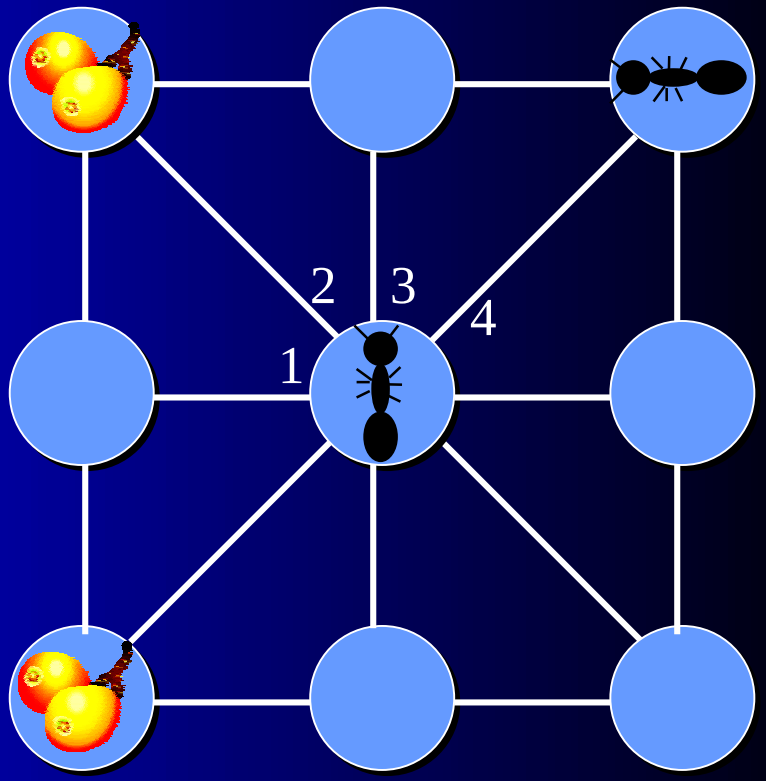
- Genetic Algorithm -

Goal:

To find the genetic code of the most efficient forager

Ant behavior:

- follow the strongest pheromone,
- occasionally change direction,
- pick up food,
- emit pheromone,
- carry food to nest,
- drop food on route to nest



Ant Farm

- MPI -

```
for each simulation time {
  for each cell {
    while (ant = cell.get( )) {
            ant->behavior( );
            cpu[ant->next].put( ant );
    }
    if (cell is on CPU border)
            cpu[cell.neighbor].put( cell.pheromone );
  }
  for each neighboring cpu
    exchange ants and pheromones with cpu;
}
```

Absolute cell addressing

Separation of behavior
Simulation and migration

Ant Farm

- M++ -

```
thread ant {
private:
  inline void behavior( );
  main( ) {
    for each simulation time {
          behavior( );
          hopAlong( link );
          synchronize( );
    }
  }
}
```

Behavioral autonomy

Complete encapsulation

Relative node access
(No modification upon node re-mapping)

Programming an agent from its view point

Codi-1bit

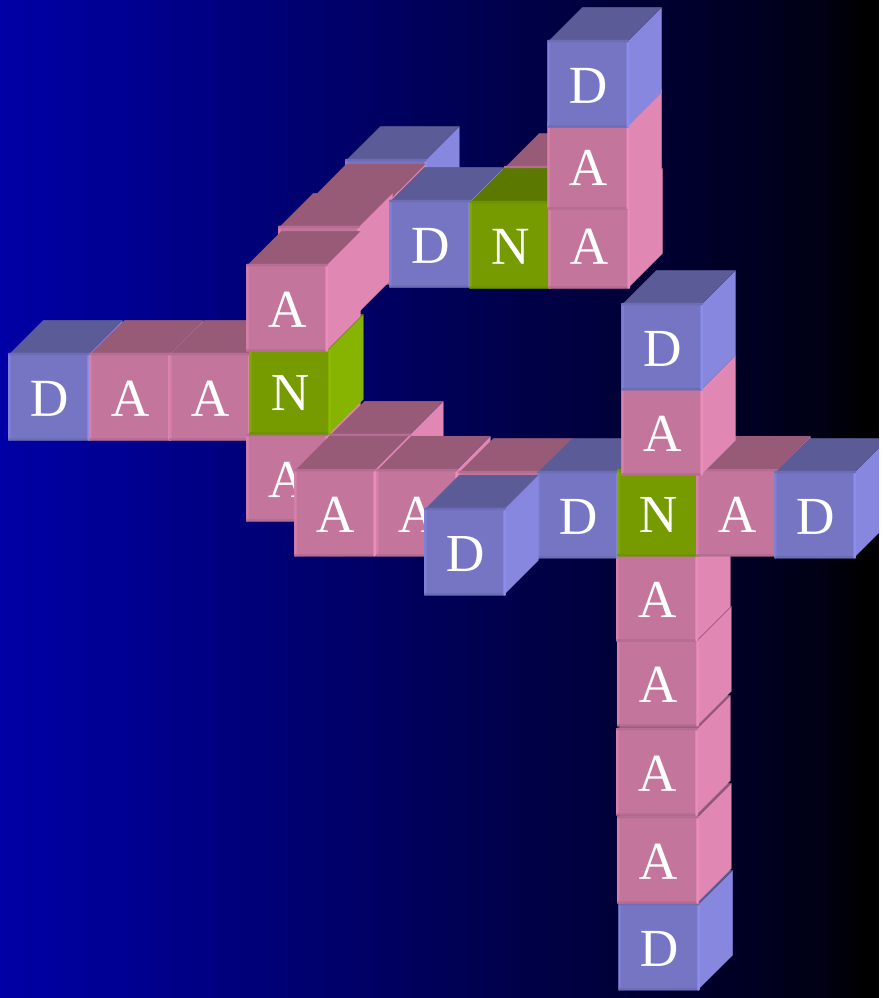
- Algorithm -

Goal:

To obtain the neural network that emits the signal along the sin θ curve

Neuron seed behavior:

- extend axons or dendrites (100 cycles)
- propagates the neural signal along the network (330 cycles)



Codi-1bit

- MPI -

```
for each growth cycle {
  for each cell {
    if (cell is in my range) {
      if (cell.type == BLANK)
        if (its chromosome is NEURON)
            cell.type = NEURON;
        else if (its 6 incoming signals result in AXON or DEND)
            cell.type = AXON or DEND;
      }
      for each of its 6 outgoing signal
            compute the signal type;
  }}
  inter-processor signal exchange;
}
```

Cell size and type is fixed

Meaningless signals propagated to cells that have become a part of network

Codi-1bit

- M++ -

```
thread neuron {
private:
  main( ) {
    for each growth cycle {
      if (cycle is 0)
        create node<Neuron>
      for each of 6 directions {
        if (node<Base>.state( ) == AXON or DEND)
              create node<Axon or Dend>….;
        else
              continue;
        create link<NeuralLink>….;
        if (forkAlong( ) == CHILD) break;
      }
      if (I'm a parent) break;
} } }
```

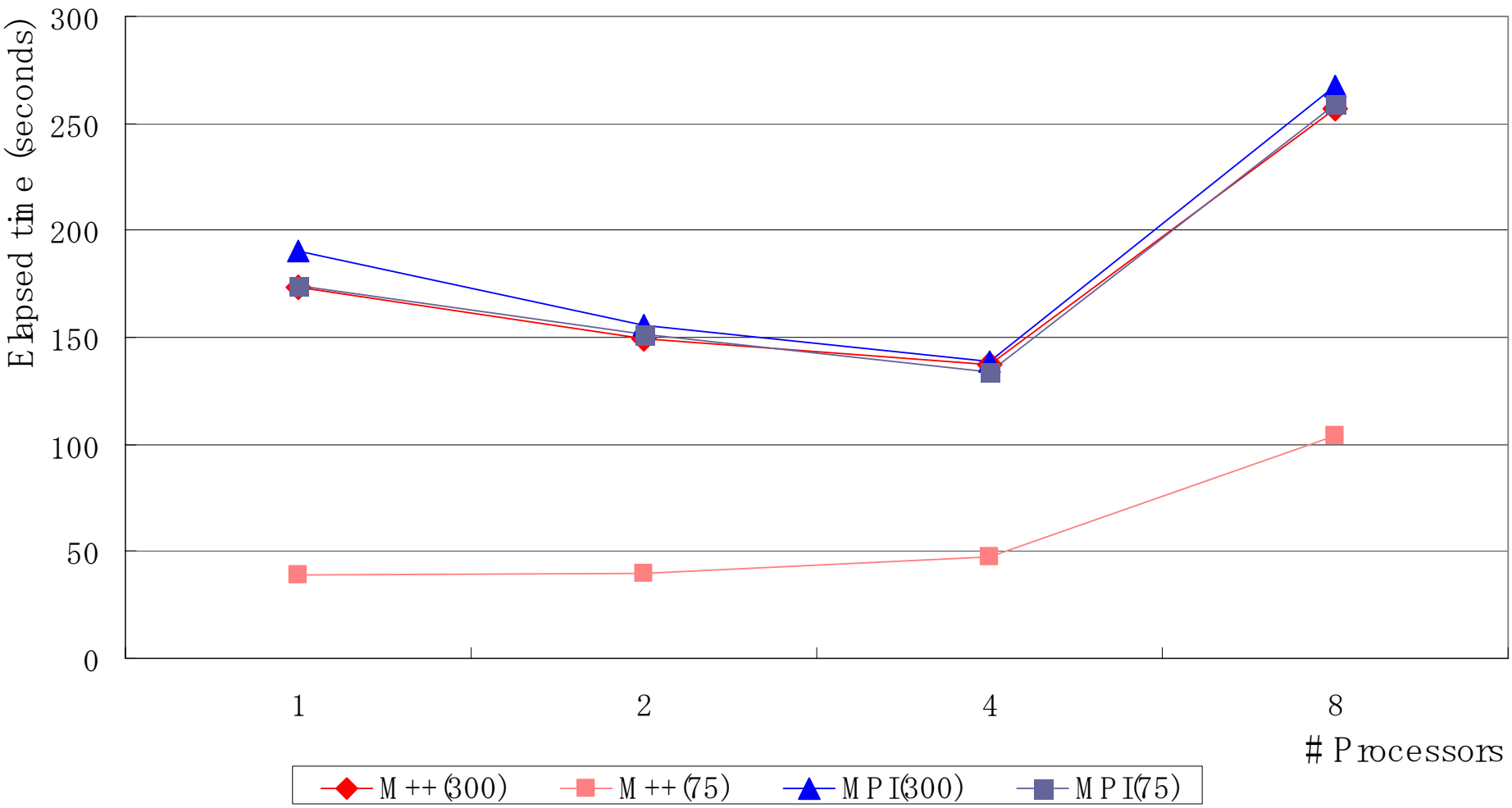
Dynamic creation of neural network

Polymorphous nodes

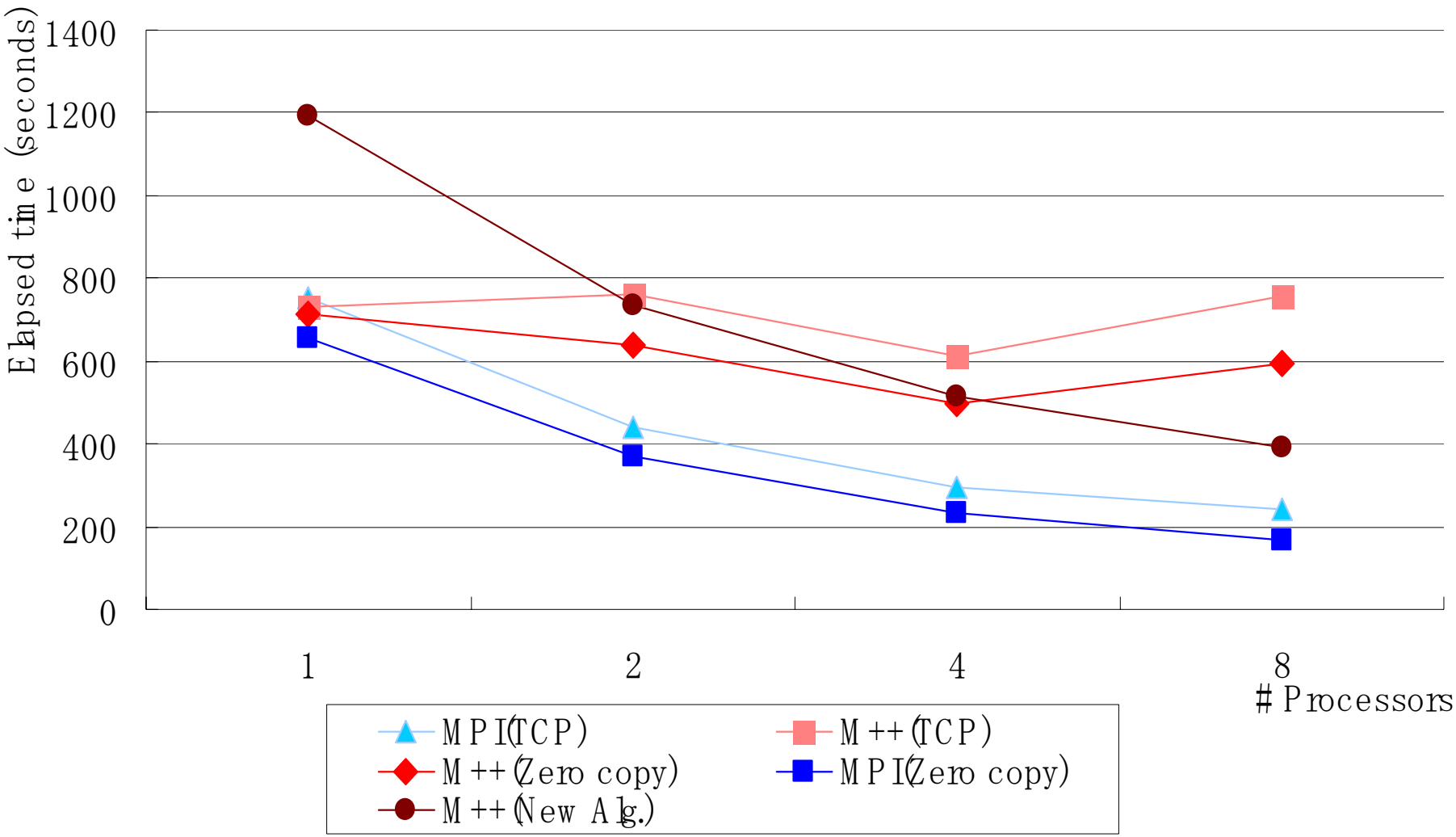
Neuron threads work on the tip of each branch.

Construct flexible network and narrow semantics gap

Ant Farm


Elapsed time (seconds)
300
250
200
150
100
50
0
1
2
4
8
Processors
M++(300)
M++(75)
MPI(300)
MPI(75)

Codi-1bit


Elapsed time (seconds)
1400
1200
1000
800
600
400
200
0
1
2
4
8
# Processors
MPI(TCP)
M++(TCP)
M++(Zero copy)
MPI(Zero copy)
M++(New Alg.)