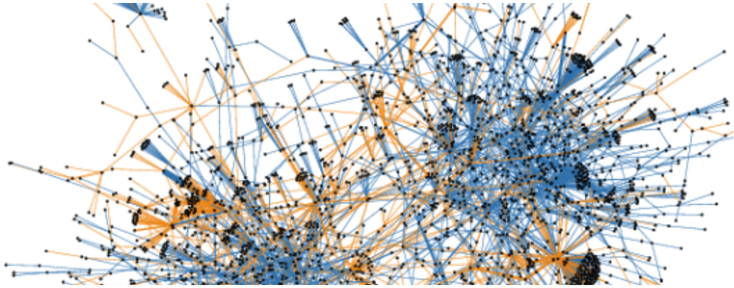
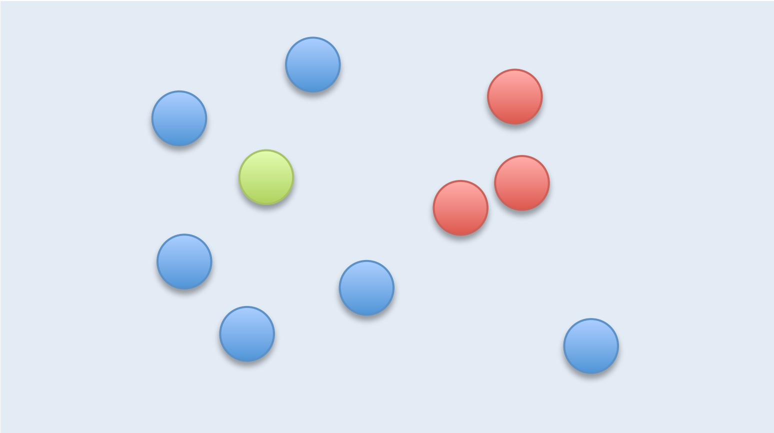


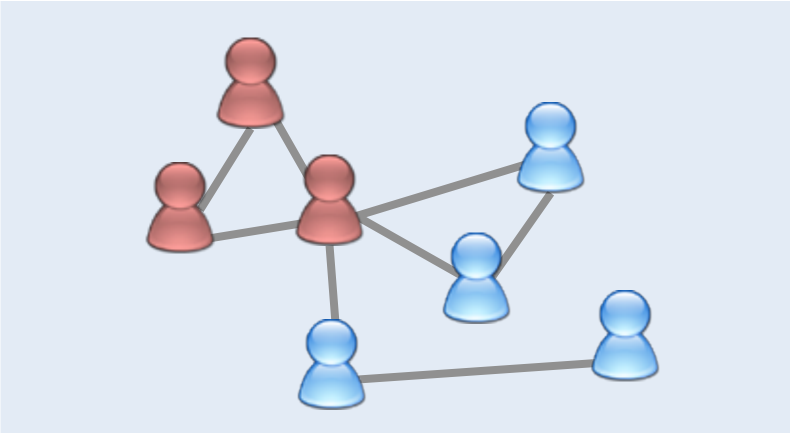
Agent-based Model Simulation with Twister

Bingjing Zhang, zhangbj@indiana.edu
Lilian Weng, weng@indiana.edu

Agent-Based Modeling

- What is *Agent-Based Modeling* (ABM)[1]?
  - A powerful simulation technique
  - Manipulations of agents on
    - A plane with randomly scattered points
    - A network/graph





[1] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences, 99:7280–7287, 2002.

Agent-Based Modeling

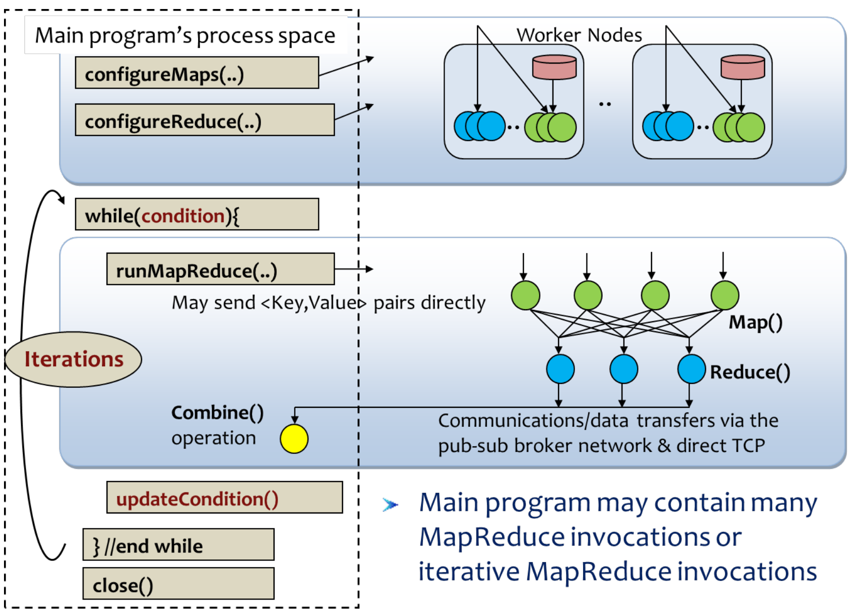
- Applications:
  - Epidemiology, spreading of diseases
  - Social networks
  - Consumer behavior in business scenario

Current Problems of ABM Simulation

- Multiple sequential iterations
- Hard to do a graph partition
- Communication-intensive
- It is easy to run a simulation on a single machine, but it is too slow

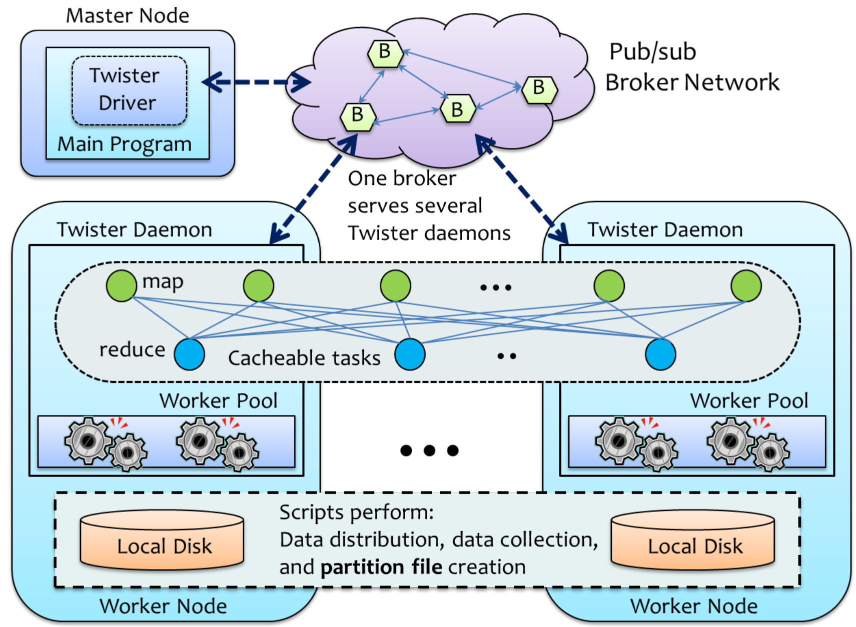
Twister
Iterative MapReduce

Iterative Programming Model



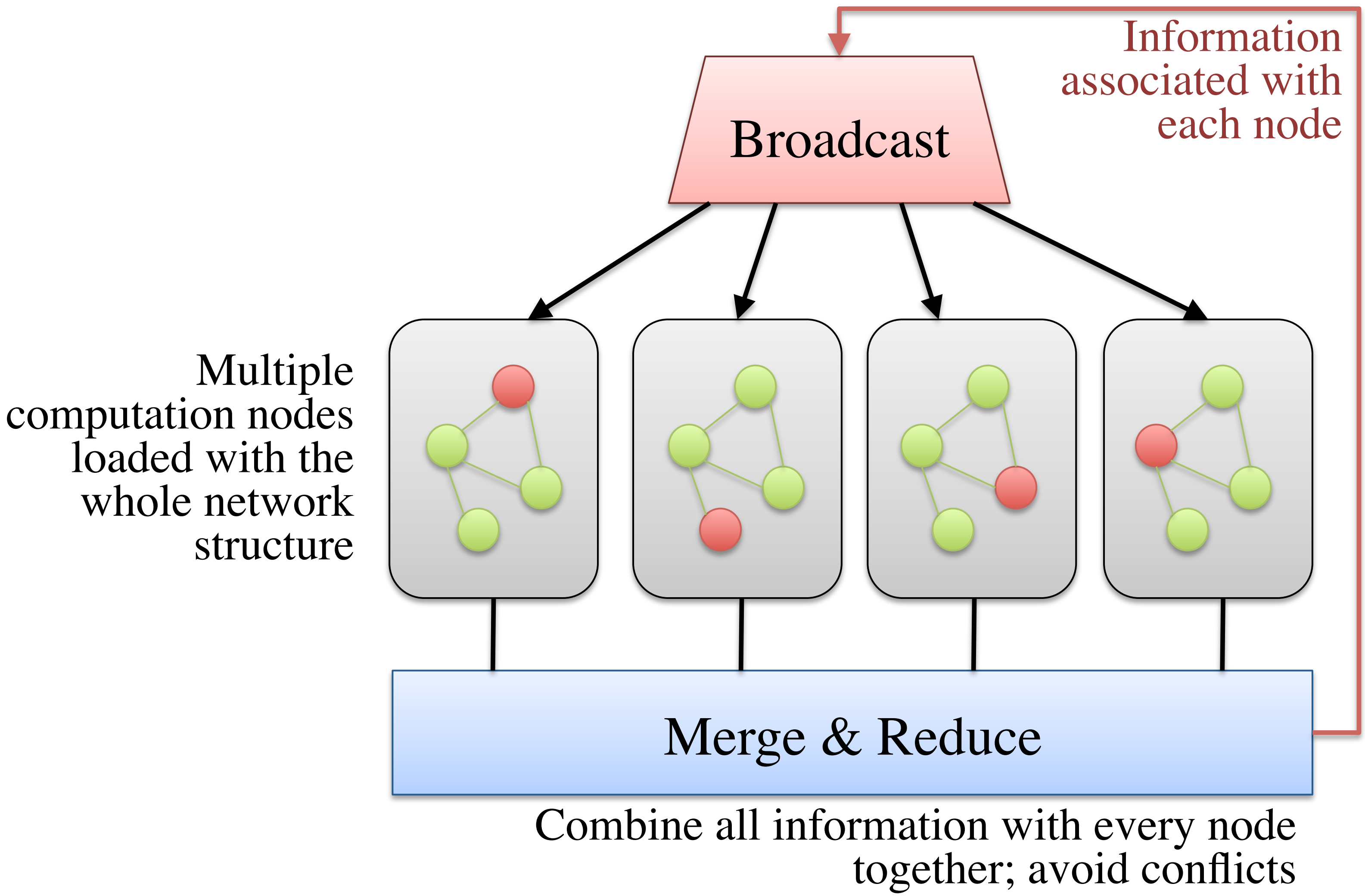
- Main program may contain many MapReduce invocations or iterative MapReduce invocations

Architecture



[2] J.Ekanayake, et al., Twister: A Runtime for iterative MapReduce, in Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010 conference June 20-25, 2010. 2010, ACM: Chicago, Illinois.

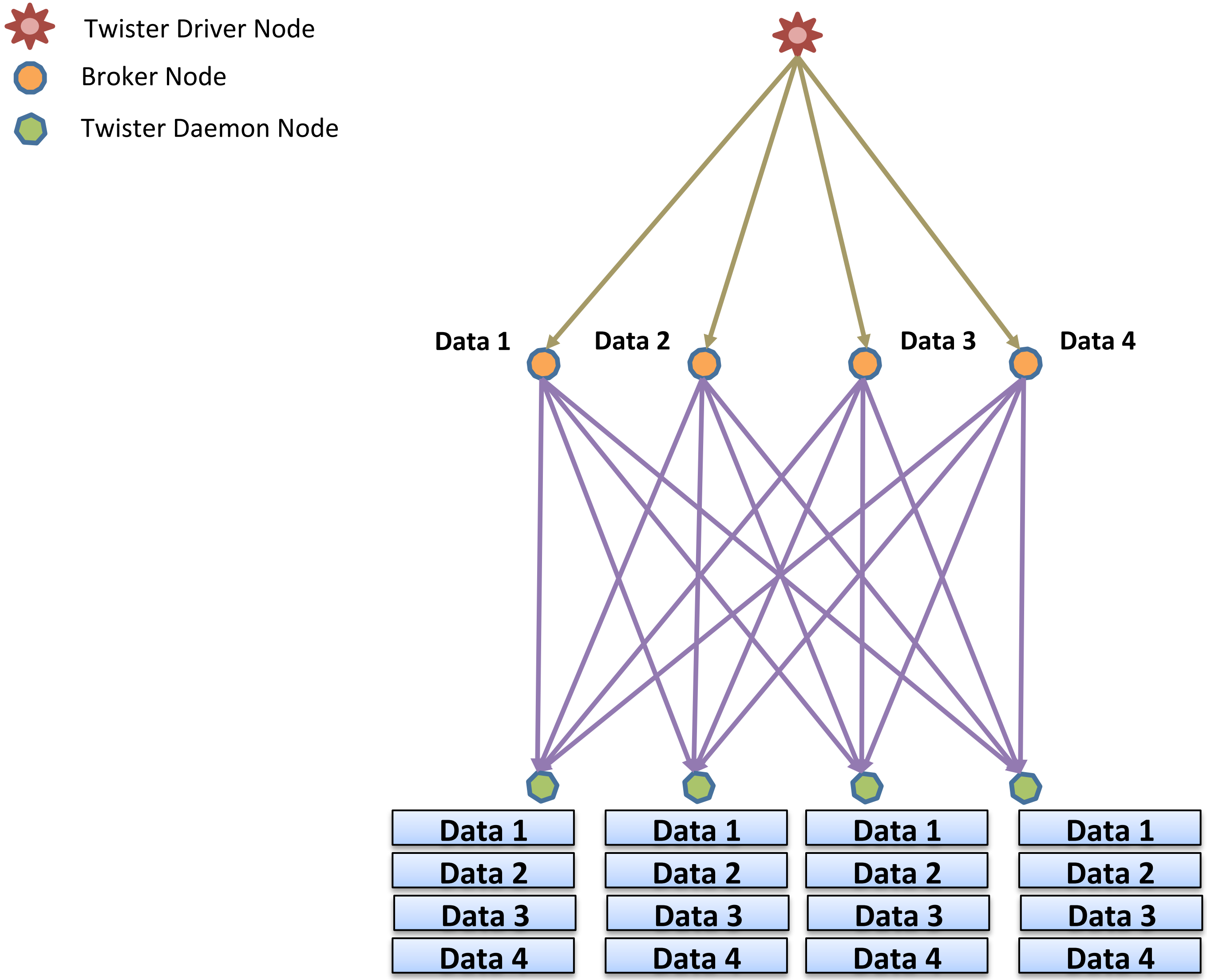
A Possible Solution of Running Simulation on Twister



Optimization on Twister

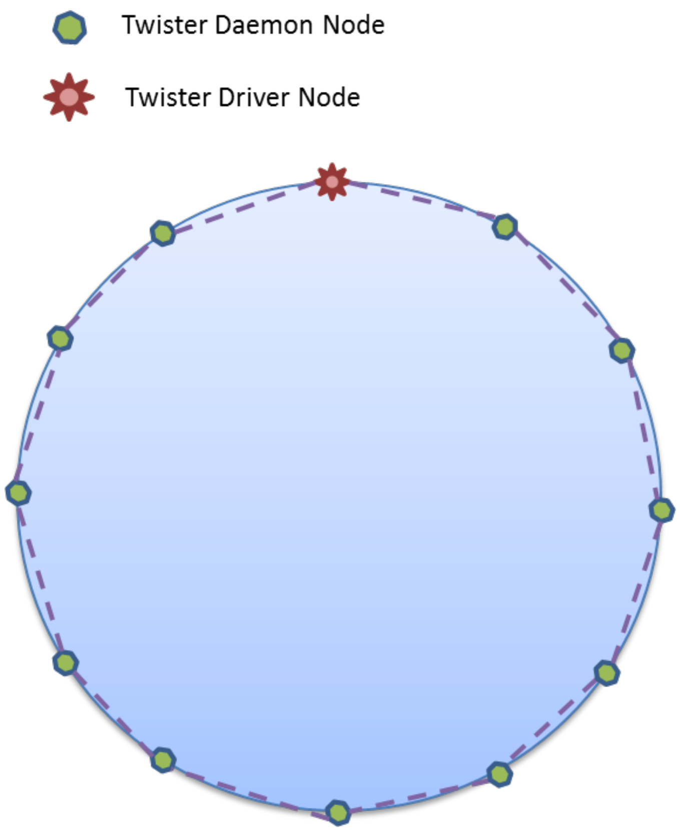
- Broadcasting
  - All-to-all Broadcasting
  - Chain Broadcasting
  - Minimum-spanning tree algorithm
- Merge and Reduce
  - Merge Operation
  - Reduce Scatter

All-to-all Broadcasting

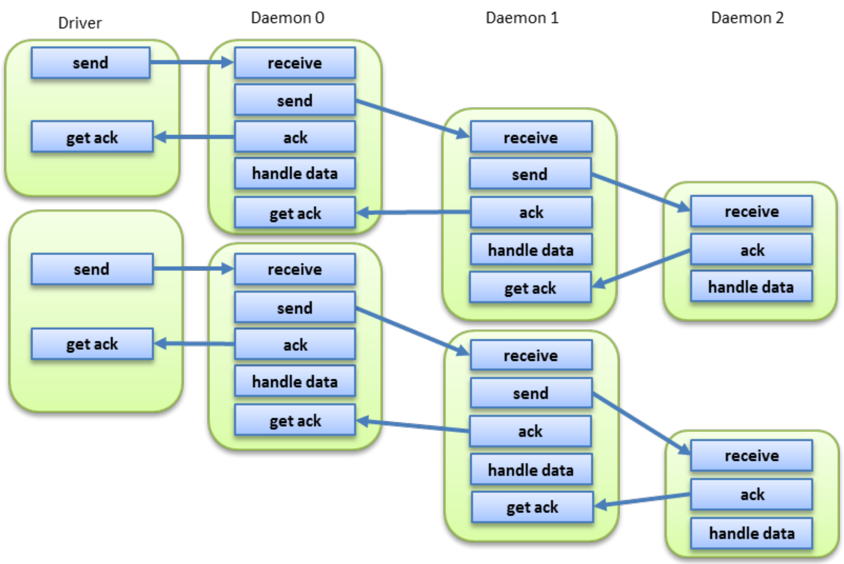


Chain Broadcasting

Chain Topology



- Every daemon gets the partial data from the socket buffer and forward them to next daemon
- Parallel data sending by pipeline



Minimum-spanning Tree Broadcasting

Hypecube

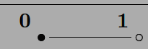
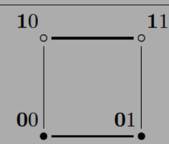
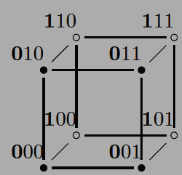
d	p	comment	shape
0	1	A single node.	•
1	2	Duplicate the 0 dimensional cube, and connect corresponding nodes with a link.	
2	2^2	Duplicate the 1 dimensional cube, and connect corresponding nodes with a link.	
3	2^3	Duplicate the 2 dimensional cube, and connect corresponding nodes with a link.	
⋮	⋮	⋮	⋮
d	2^d	Duplicate the $d-1$ dimensional cube, and connect corresponding nodes with a link.	⋮

Figure 2: Construction of hypercubes.

Mechanism

Node 0	Node 1	Node 2	Node 3
	x_0		
	x_1		
	x_2		
	x_3		

Node 0	Node 1	Node 2	Node 3
	$x_0 \rightarrow$		
	$x_1 \rightarrow$		
	$x_2 \rightarrow$		
	$x_3 \rightarrow$		

Step 1

Node 0	Node 1	Node 2	Node 3
	$\leftarrow x_0$	$x_0 \rightarrow$	
	$\leftarrow x_1$	$x_1 \rightarrow$	
	$\leftarrow x_2$	$x_2 \rightarrow$	
	$\leftarrow x_3$	$x_3 \rightarrow$	

Step 2

Node 0	Node 1	Node 2	Node 3
x_0	x_0	x_0	x_0
x_1	x_1	x_1	x_1
x_2	x_2	x_2	x_2
x_3	x_3	x_3	x_3

Figure 4: Minimum-spanning tree algorithm for broadcast.

[3] Ernie Chan et, al. Collective Communication: Theory, Practice, and Experience. Concurrency and Computation: Practice and Experience. 2007

Merge

- We can do merge on partial collection of the KeyValue pairs if

- $collect(kv_1, kv_2, kv_3, kv_4, \cdots)$

$$= collect(kv_1) \oplus collect(kv_2) \oplus collect(kv_3) \oplus collect(kv_4) \cdots$$

  - *collect* is the collect operation on KeyValue pairs.
  - $\oplus$ is the conjunction of collect operation.

Reduce: Recursive-halving for Reduce-scatter

Node 0	Node 1	Node 2	Node 3
$x_0^{(0)}$	$x_0^{(1)}$	$x_0^{(2)}$	$x_0^{(3)}$
$x_1^{(0)}$	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$
$x_2^{(0)}$	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$
$x_3^{(0)}$	$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$

Node 0	Node 1	Node 2	Node 3
$x_0^{(0)}$	$x_0^{(1)}$	$x_0^{(2)}$	$x_0^{(3)}$
$x_1^{(0)}$	$x_1^{(1)}$	$x_1^{(2)}$	$x_1^{(3)}$
$x_2^{(0)}$	$x_2^{(1)}$	$x_2^{(2)}$	$x_2^{(3)}$
$x_3^{(0)}$	$x_3^{(1)}$	$x_3^{(2)}$	$x_3^{(3)}$

Step 1

Node 0	Node 1	Node 2	Node 3
$x_0^{(0:2:2)}$	$x_0^{(1:3:2)}$		
$x_1^{(0:2:2)}$	$x_1^{(1:3:2)}$		
		$x_2^{(0:2:2)}$	$x_2^{(1:3:2)}$
		$x_3^{(0:2:2)}$	$x_3^{(1:3:2)}$

Step 2

Node 0	Node 1	Node 2	Node 3
$x_0^{(0:3)}$			
	$x_1^{(0:3)}$		
		$x_2^{(0:3)}$	
			$x_3^{(0:3)}$

Figure 9: Recursive-halving algorithm for reduce-scatter. Notation: $x_i^{(j_0:j_1:s)} = \sum_j x_i^j$ where $j \in \{j_0, j_0 + s, \cdots, j_1\}$ and $x_i^{(j_0:j_1)} = \sum_j x_i^j$ where $j \in \{j_0, j_0 + 1, \cdots, j_1\}$.

Problems

- Twister may not be a perfect solution for agent-based model simulation, but it is definitely can help improve the performance compared to the one on a single machine.

- Speed up will not be a good measure of improvement, because the simulation is communication-intensive

- Large cluster and memory can be a good bonus.

Project Timeline

Time	Tasks
11/01-11/15	Architecture buildup and preliminary implementation of the experiment
11/16-11/23	Implementation optimization
11/24-12/02	Performance testing and parameter tuning
12/03-12/10	Final report