MPI simulation of a MIMO node choosing algorithm

CSE 633 Project Presentation

Xiaoxing Yu

Instructor: Russ Miller

Dec.09 2010 Fall semester

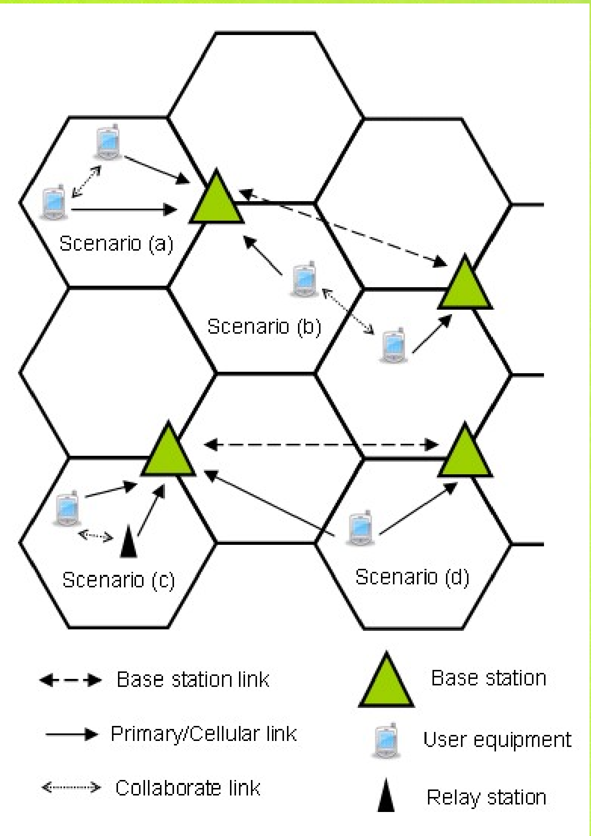
CSE department of SUNY at Buffalo

Background

What's MIMO?

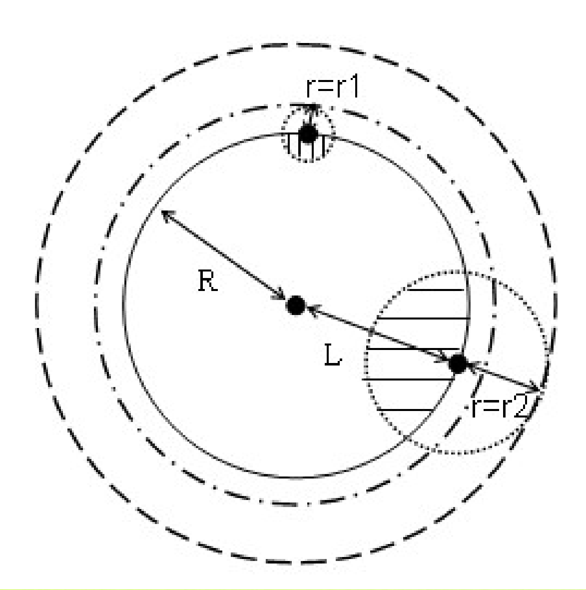
What's MIMO collaboration?

Why MIMO collaboration is good?



Proof of effectiveness

- The solid inner circle is the transmission range;
- The middle dotted circle is the overall possible range of walking neighbor node;
- The outer dotted circle is the overall possible range of vehicle neighbor node;
- Black dots are request node, single edge vehicle and walking helping neighbor (with their moving ranges).



To compute $S_{overlap}$, we set:

$$\theta 1 = \arccos[(L^2+R^2-r^2)/(2*L*R)];$$
$$\theta 2 = \arccos[(L^2+r^2-R^2)/(2*L*r)];$$

then,

$$S_{overlap} = (\pi*R^2*2*\theta 1/360^\circ) - (R^2*\sin\theta 1*\cos\theta 1) + (\pi*r^2*2*\theta 2/360^\circ) - (r^2*\sin\theta 2*\cos\theta 2)$$

While $R-r < L \leqslant R$;

The single node coverage possibility can be computed as:

$$p = S_{overlap} / (\pi*r^2).$$

Total coverage possibility is:

$$P = (1-(1-p)^N)$$

Process

Firstly UE will perceive low signal power from antenna or bandwidth assignment by accessing information exchange with BS. Then, UE can choose to switch to our MIMO mode becoming request node. It then broadcasts *hello message*. The idle neighbor in transmission range will answer request node with an *ACK message*.

After gathering *ACK messages* in a time limit, assuming M messages received, request node will compute "valid value" V of each response by evaluation formula V= α*sp1+β*sp2, while α and β are two optimized parameters that can be pre-set in UE. This V is general assessing of stability, distance and possible quality improving of helping nodes.

Then request node will choose at most N (pre-set) appropriate nodes among those M nodes as MIMO collaborate nodes by "valid value" V, and store them into his forwarding table T1. The number of ultimate chosen nodes in T1 is k. T1 can be seen as routing table and reference for MIMO downlink data merging process.

Algorithm and pseudo-code

```
Collaborate choosing ()
{ if (M == 0){ exit(0);} // no response to hello, then fail
  for (i=0; i<M; i++) // has responses, compute each V
  {compute V for each item by V[i]=α*sp1[i]+β*sp2[i]}
  list by decreasing V[i];
  if (M <= N) {copy all helping ID into T1} // choose M nodes if M<=N
  else {copy top N items into T1} // choose N nodes if M>N
  send Confirmation message to k collaboration nodes;}

Route maintain on request node()
{ while(MIMO mode)
  {while(T1 not empty) // keep working until T1 has no item
   {for (i=0; i<k; i++) //delete down link from T1 and refresh k
     { if (node[i] link down)
       { delete node[i] from T1;
          k--;}}}}}

Route maintain on helping node()
{ if (Request Node link down)  // if link down
  {Request Node=null;  //delete Request Node, end collaborate
    end collaborate;}}
```

Mobility Simulation

We value the nodes' moving behaviors by transferring the attained energy information into distance measurement.

While all distances can be computed and approximately simulated by the ideal moving formula in Physics:

$$S = d_0+v_0*t+a_0*t^2$$

For each node, there are two moving behaviors we interested, one against the request node, and the another against the base station. So there are two distance here we want to compute for each node, S_1 and S_2 .

Why Parallel

- **Massive data and calculation. Complexity may be between O(nlogn) and O(n^2), depended on the node accessing situation.**

- **Shorter running time and larger computing capability. The running time of sequential program is about 30 seconds for 5 nodes on my own laptop MATLAB.**

- **Independent data relation and computing processes for each node due to independent nodes' information.**

How to parallel

I intend to simulate the scenario with a amount of helping nodes to statistic the algorithm efficiency, which requires huge computation of V value, and the sorting algorithm also needs to enhance the computing speed.

Nodes' computation separate into groups of "computing nodes" on "edge"

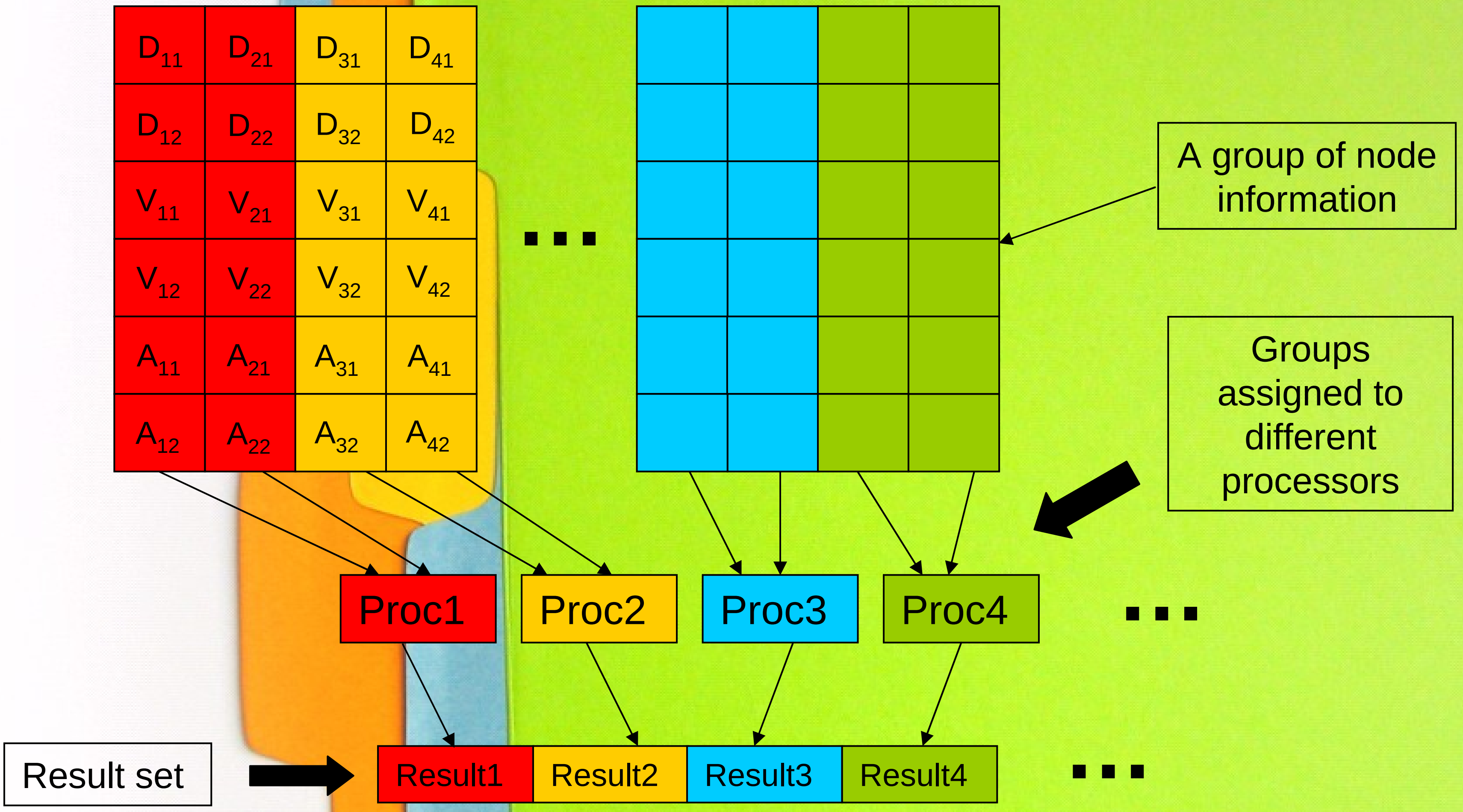
4~32

Returning results back to a certain set

Separate set again to "computing nodes" on "edge" to do faster sorting

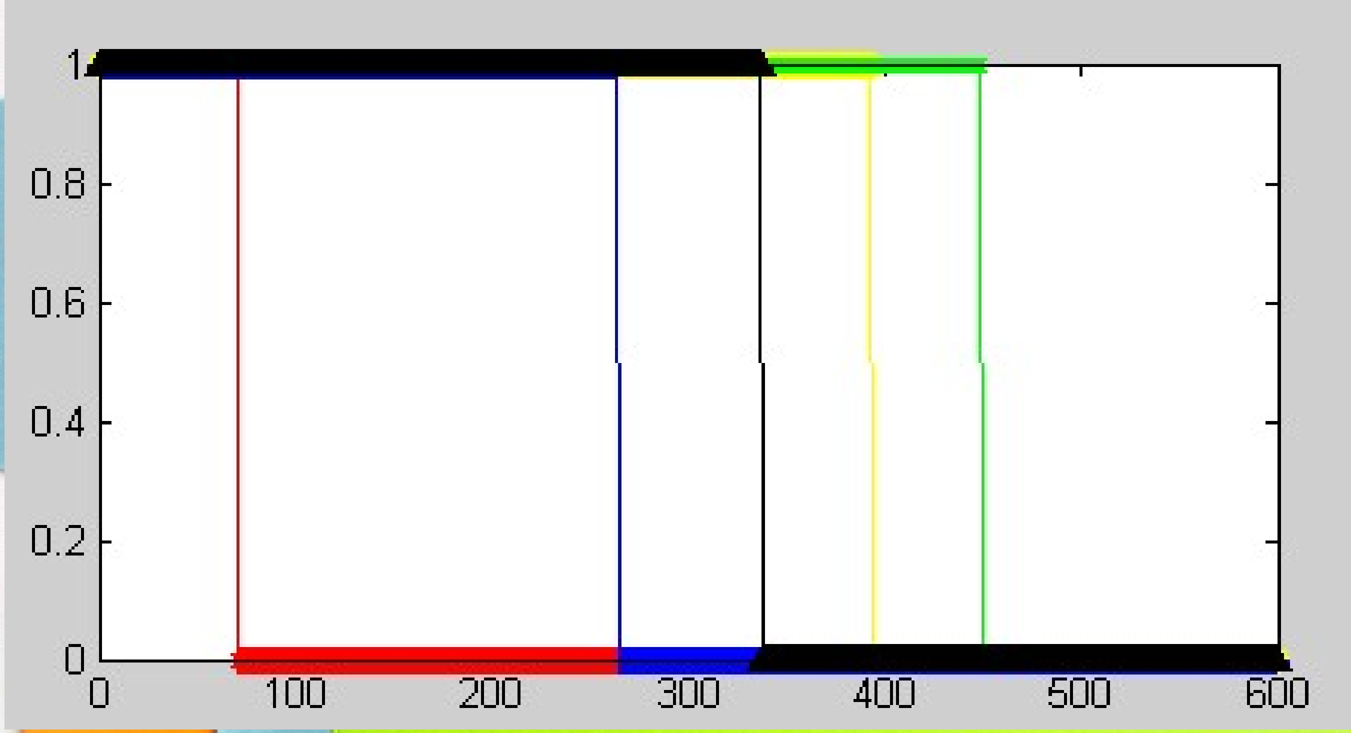
4~32

Structure



D11 D21 D31 D41
D12 D22 D32 D42
V11 V21 V31 V41
V12 V22 V32 V42
A11 A21 A31 A41
A12 A22 A32 A42
...
A group of node information
Groups assigned to different processors
Proc1
Proc2
Proc3
Proc4
...
Result set
Result1
Result2
Result3
Result4
...

Simulation Results



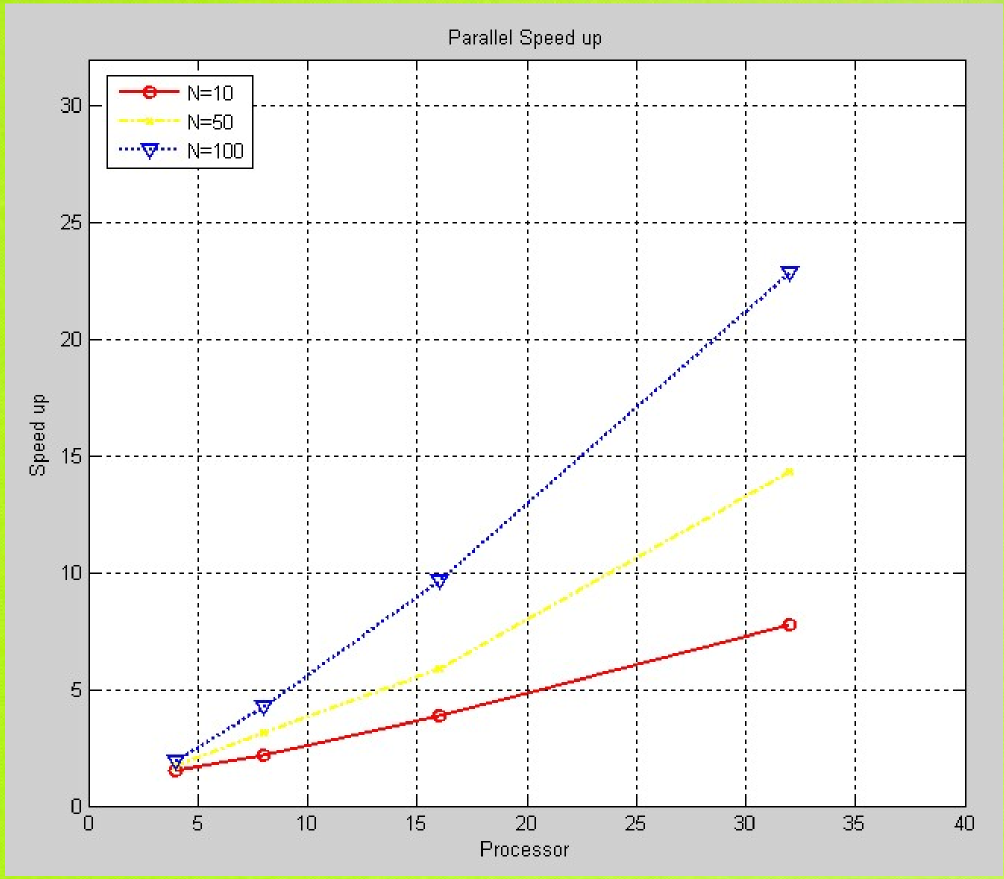
Service holding time

(when choosing 5 nodes, “1” for “hold” and “0” for “drop”)

Simulation Results (cont)

Here is the speed up chart applying MPI programming

- **Obvious speed up trends accompanied with the growing of the number of processors**
- **Potential massage delay effects the running time due to massive data passing**



Conclusion and Future work

This project achieves the task for being an algorithm simulation. It accomplishes part of the theoretical proof of the MIMO collaborative.

For being this parallel computing project, it successfully applies parallel programming into practical work, and proves the powerful capability and computing speed advantages of parallel computing.

The future work of this project may focus on:

- **Adaptive evaluating parameters, and merge them into the new parallel computing process.**
- **Refining the program performance about less resources and faster speed.**

Reference

- **CCR parallel programming guidance:**

https://www.ccr.buffalo.edu/display/WEB/Training

- **M. Karakayali, G. Foschini, and R.Valenzuela, “Network coordination for spectrally efficient communications in cellular systems”, Wireless Communications, vol.13, no.4, pp. 56–61, Aug. 2006.**

Thanks

Questions?