

Simulated Annealing to Optimize Layout on BG/L

Gyan Bhanot
IBM Research

Topology Matters

- Test on 128 Way BGL
- MPI_ISEND,
MPI_IRECV,
MPI_WAIT
- Machine is (8,4,4)
- Hops = 6
- Cost/Time = 6 x

Data	Near node	Far node
0.2MB	0.003s	0.019s
0.4MB	0.006s	0.044s
2.0MB	0.03s	0.20s
4.0MB	0.06s	0.40s

Simulated Annealing Algorithm

- $F = \sum C(i,j)H(i,j) = \text{Cost Function}$
- Start with some mapping M_0 of variables to BGL nodes
- Compute $F(0)$ of mapping
- Swap/move variables on nodes (keeping compute load balance) to get mapping M_1
- Compute $F(1)$
- Replace M_0 by M_1 with
$$\text{Prob} = \exp\{[F(0)-F(1)]/T\}$$
- Repeat while slowly decreasing T from initial large value in small steps until converged
- Final Map should be better than initial map ??

How would this work?

- Procedure for Run Time System
 - Code maps MPI tasks to random BG/L nodes
 - Run code for few minutes to understand communication traffic
 - Run SA Algorithm to find optimum map
 - Remap MPI Tasks using optimum map (ie. redefine MPI_COMM_WORLD)

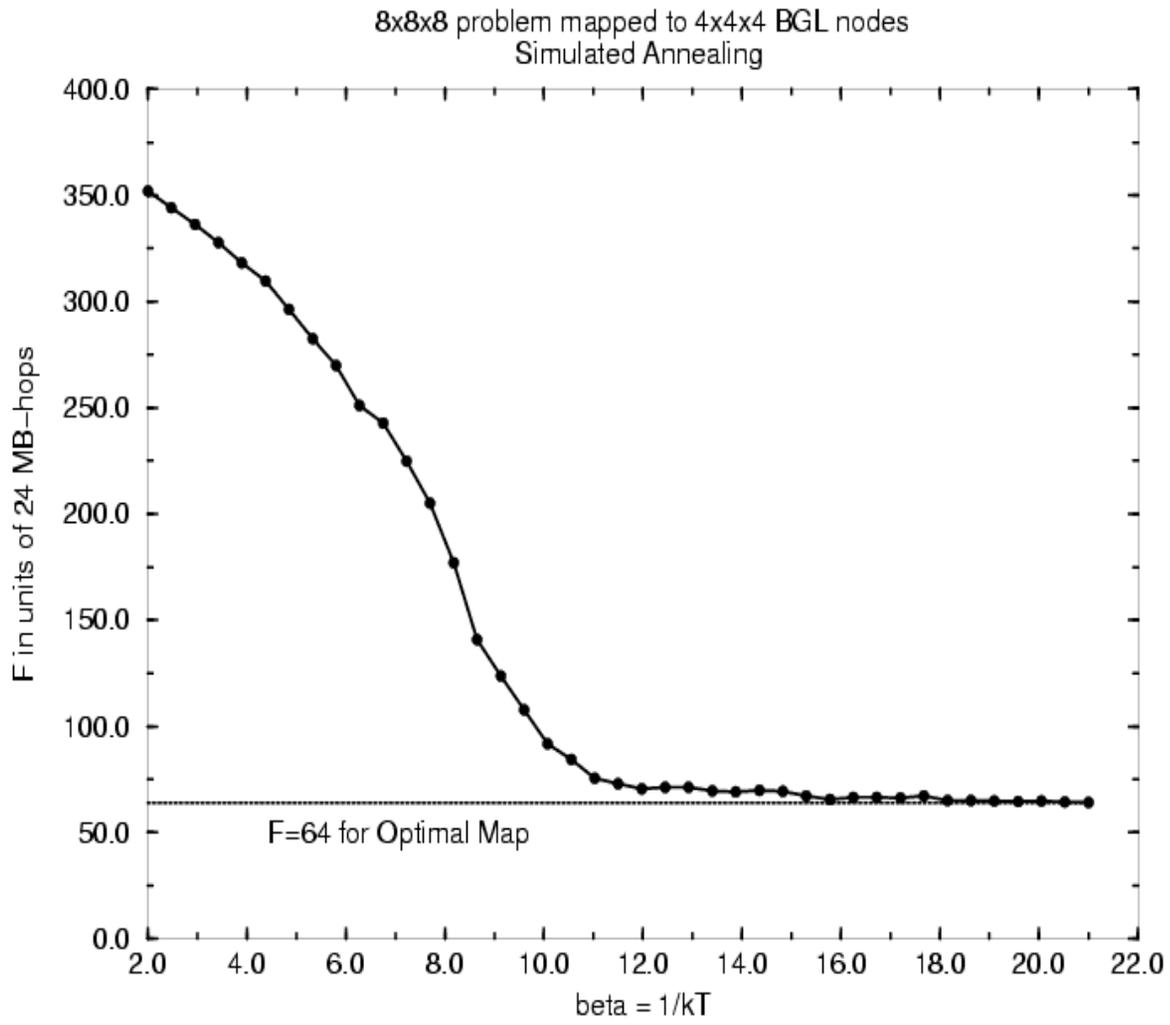


FIGURE 3

UMT2K Communication Imbalance

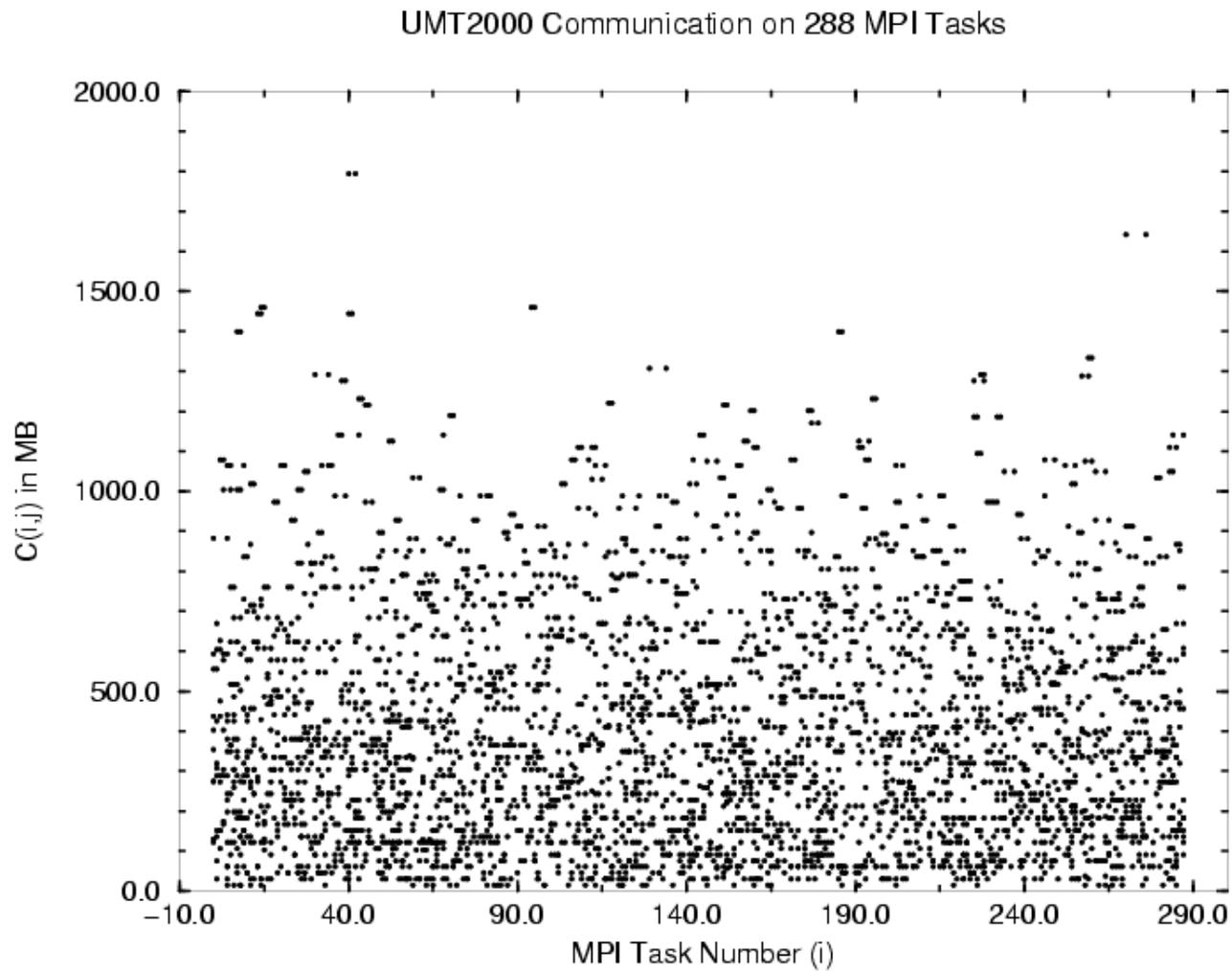


FIGURE 1

UMT2000 mapping from 288 nodes to 4x4x4 BGL
Simulated Annealing

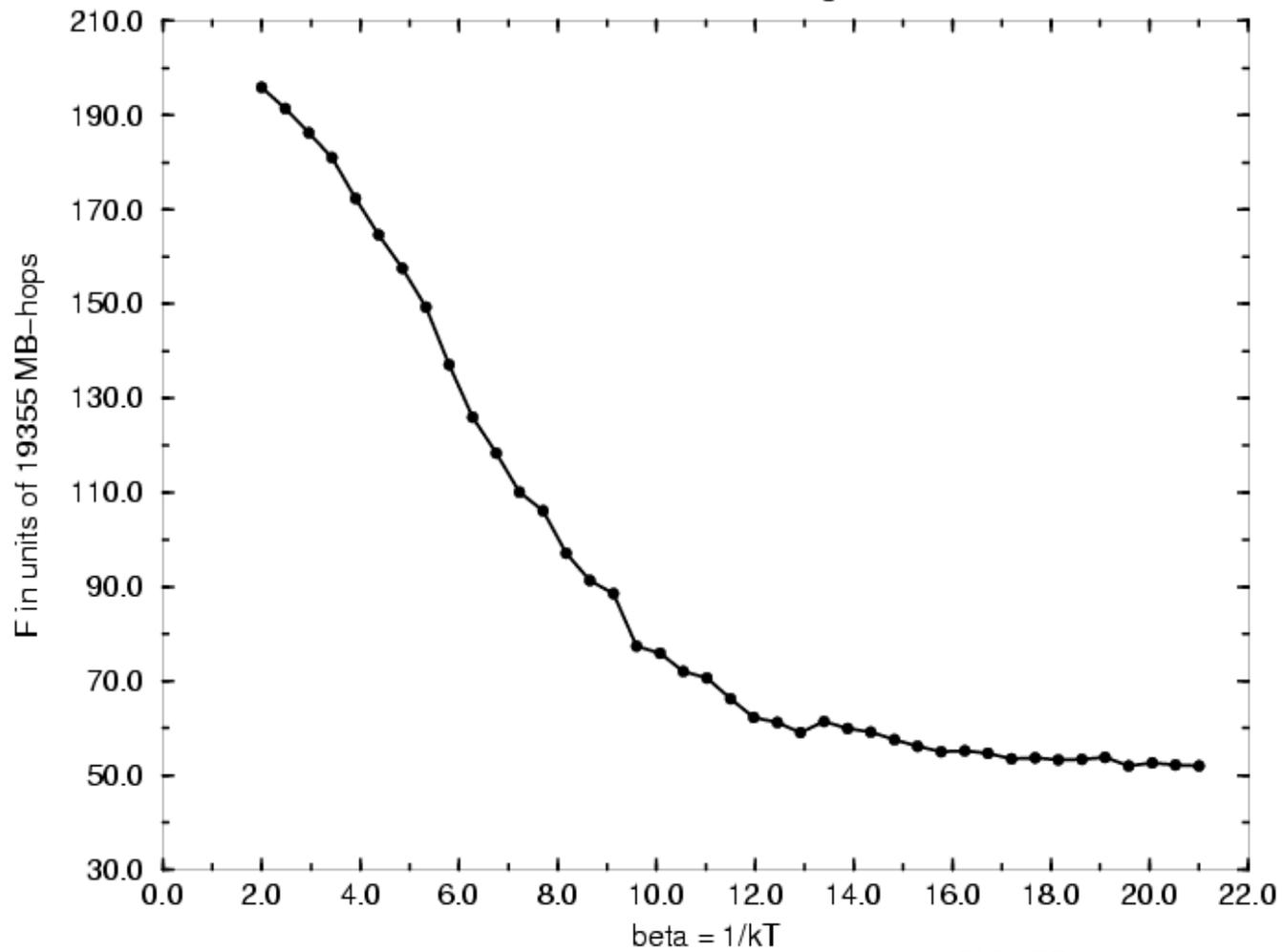
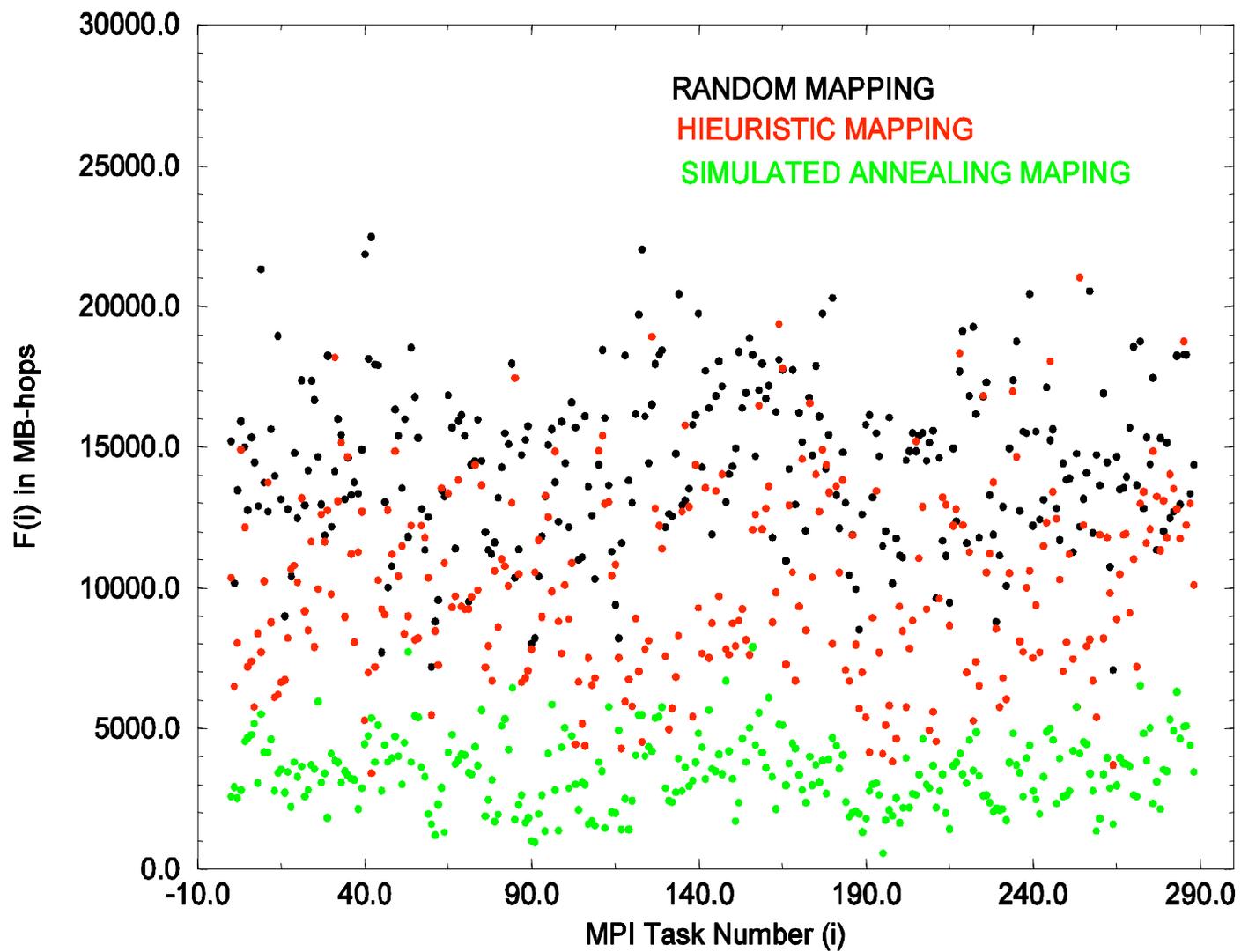
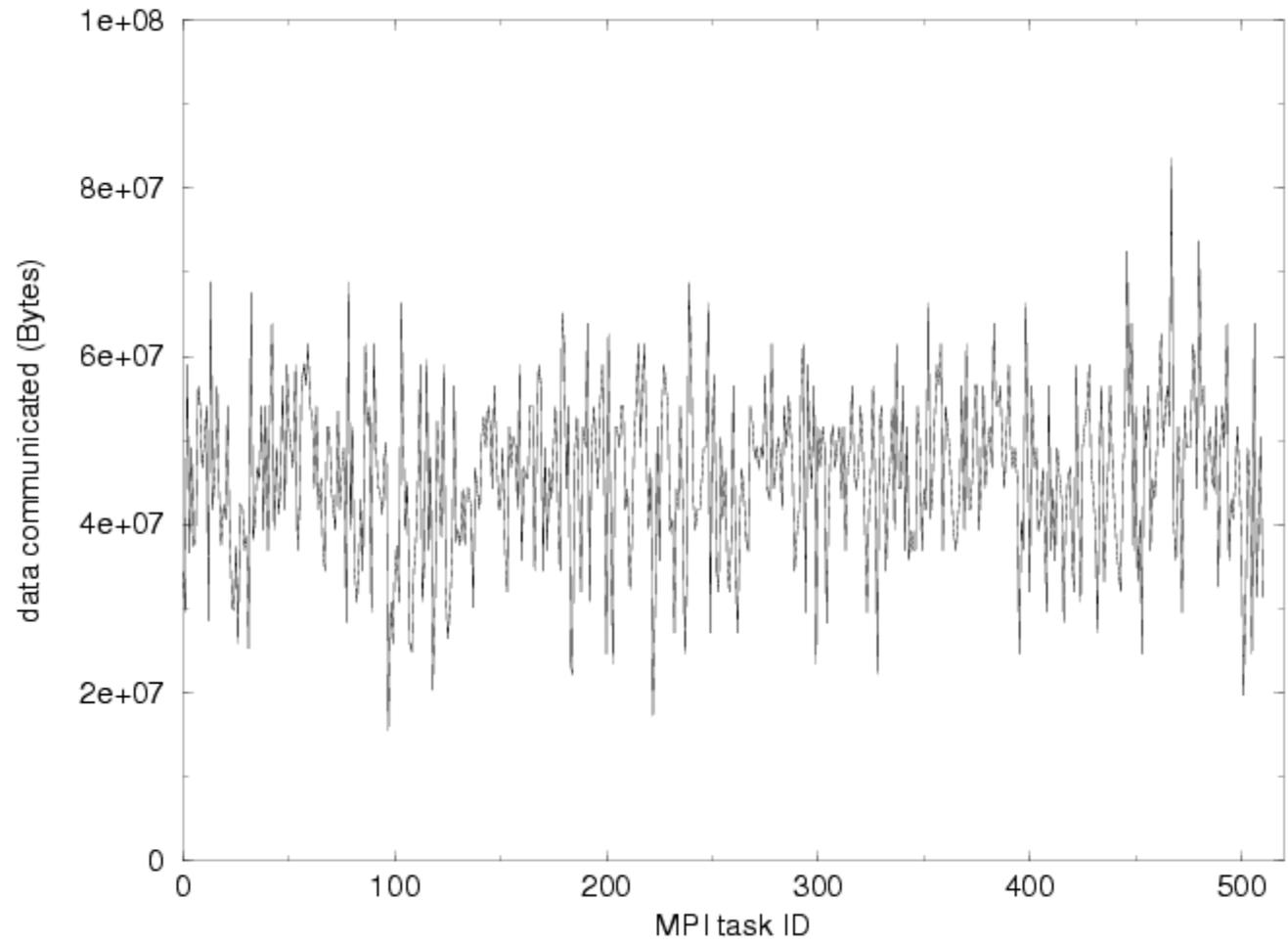


FIGURE 2

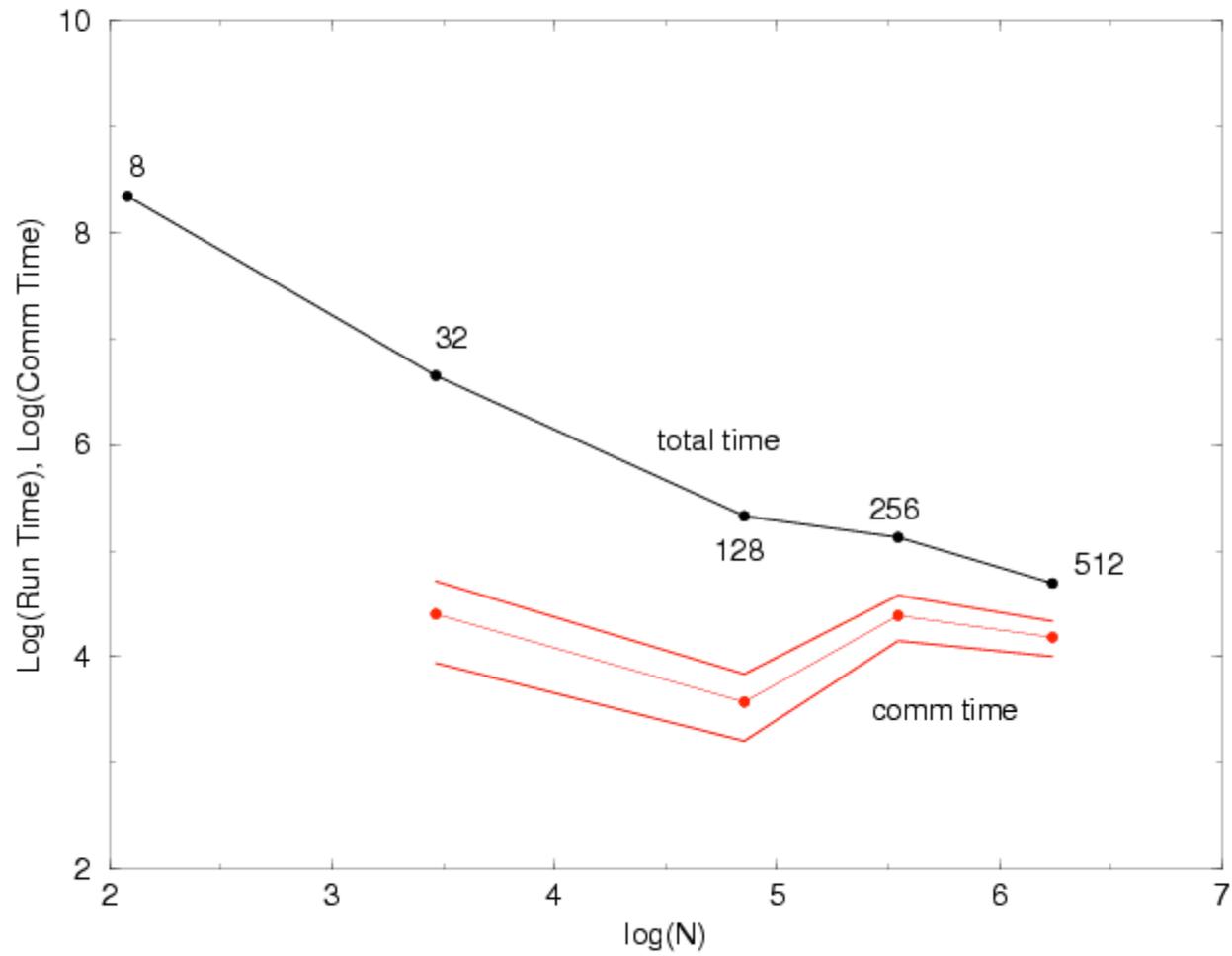
UMT2000 maps from 288 nodes to 4x4x4 BGL



UMT2K communication on 512 nodes



UMT2K Timings on BGL (Oct 12, 2003)



What needs doing?

- Find a good cost function F which accounts for multiple paths, packets, buffer sizes. Perhaps need to use direct timing of MPI functions.
- Find a good Simulated Annealing protocol
- Provide functionality to remap MPI task ID on BG/L
- Test on Applications