Ex 5 – Functional Programming in Concurrent and Distributed Systems

Ron Kotlarsky 204130538

RingA(N,M)

In this function each node spawns the next one. Once build is complete, M messages are passed. Each message is sent once the previous message has propagated through the entire ring.

Time statistics:

N	M	Time For build and Passing (Microseconds)
100	20	3429
1000	20	15814
10000	20	129733

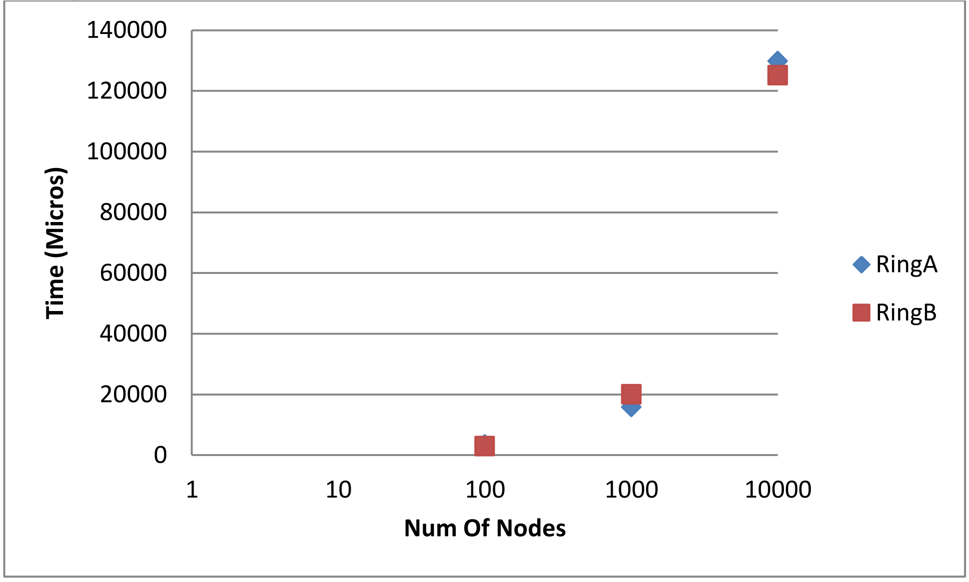
RingB(N,M)

In this function all nodes are created using a central process. Once build is complete, M messages are passed. Each message is sent once the previous message has propagated through the entire ring

Time statistics:

N	M	Time For build and Passing (Microseconds)
100	20	2911
1000	20	20045
10000	20	125185

Comparison



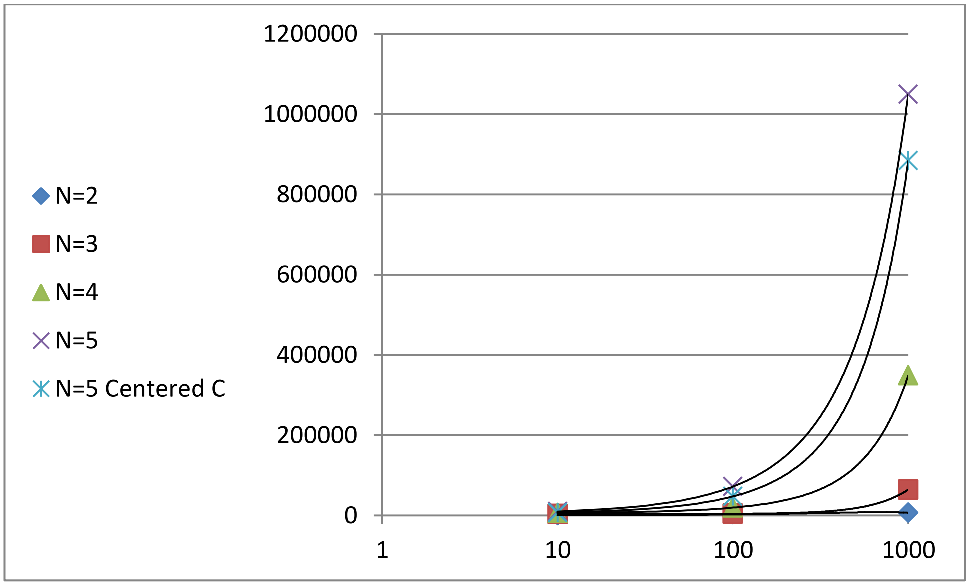
It is possible to observe that in small N's, time for creation and passing are quite similar. However in larger N's parallel creation is much faster.

Mesh(N,M,C)

In this function, nodes are created using a central process. Once all processes are spawned, the central process notifies all processes that grid was created. Once that has happened, M messages are passed from C to its neighbors and so forth. Once a new message is received, the message is stored in a map, and if it arrives again it is not passed a second time.

Process is finished once (N*N-1)*M unique ACKs are received by C.

N	M	C	Time Took to Receive All ACKs
2	10	1	848
2	100	1	2644
2	1000	1	6794
3	10	1	3724
3	100	1	4121
3	1000	1	64440
4	10	1	6399
4	100	1	19535
4	1000	1	348789
5	10	1	10050
5	100	1	72098
5	1000	1	1049336
5	10	12	7135
5	100	12	47562
5	1000	12	883921



I have chosen to test the function with C=1 for compatibility. However it is clear to see that in N=5, once C was chosen in the center of the mesh, times were reduced.