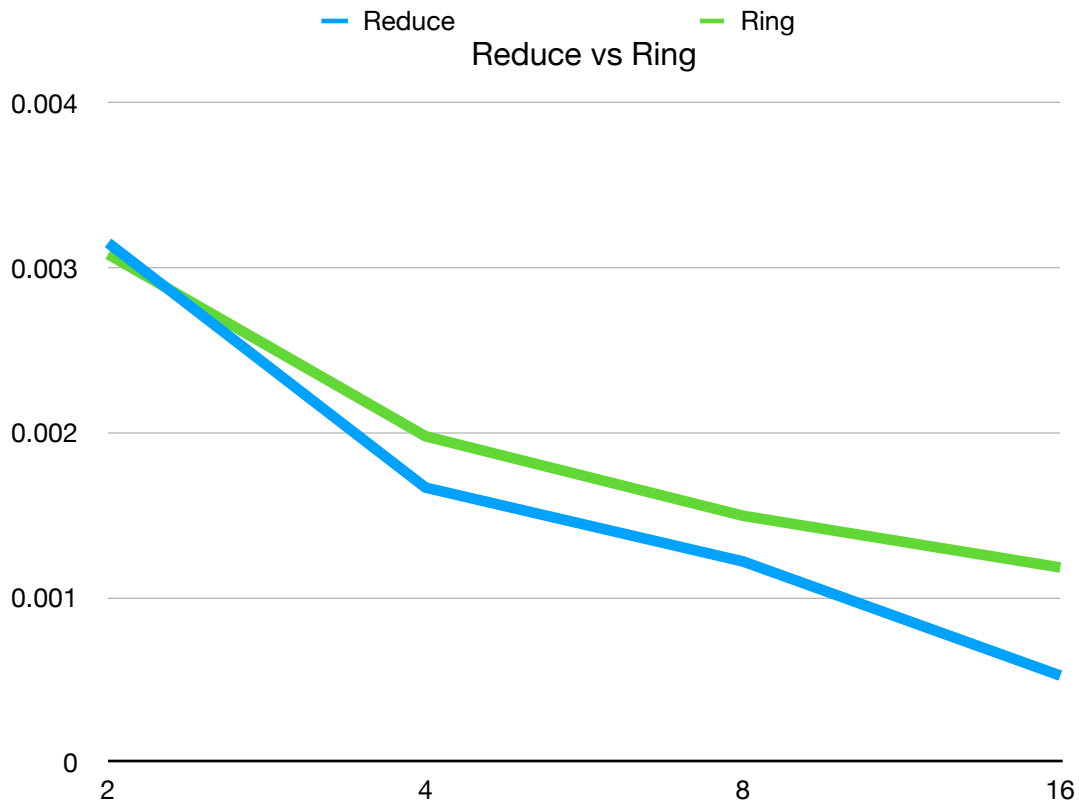
Timing Report For Reduce and Ring Topology

I - Table of timings

	Reduce	Ring
2	0.00315273	0.00308314
4	0.00166559	0.00197751
8	0.00121691	0.00149442
16	0.00052301	0.00118009



In general, the timing is decreased (faster) for both types when increasing more number of processes. However, between 2 types, the reduce timing becomes faster significantly compared to the ring timing when increasing more number of processes.

II - Questions

1. What is the topology used by MPI to implements its Reduce?

MPI_Reduce provides a way to handle the common reductions by using operation *op* parameter of predefined reduction operations (like MPI_SUM, MPI_MAX, MPI_MIN) and operate reduction on a set of elements in the input buffer of each process, and combine them into smaller set of elements, and return the output buffer of the smaller set in the rank root. It is like "many-to-one".

2. How does the ring topology affect the runtime?

Ring topology runtime depends a lot on the number of processes, or the size of the ring. As the size of the ring becomes bigger, the communication delays more. Hence, the ring topology is faster with smaller number of processes.

3. Is there a case where the ring topology may be faster?

Compared to MPI_Reduce, the only situation when ring topology may be faster is with the ring size of 2, because it only has to traverse once, which may be faster than MPI_Reduce.

4. How many messages are passed between machines when program is run with 2, 4, 8, and 16 processes?

In reduce: 2 processes: 1 messages, 4 processes: 3 messages, 8 processes: 7 messages, 16 processes: 15 messages

In ring topology: The number of messages are the same as the number of the processes. 2 processes: 2 messages, 4 processes: 4 messages, 8 processes: 8 messages, 16 processes: 16 messages