Julian Kemmerer
ECEC 414
Dr. Hempstead
Cache Coherence Experiment
*Please use 2 of my 5 late days.

Results and Conclusions

Configuration(s)

Simulations in this report varied the following parameters over the following ranges for a total of 24 simulation combinations.

- Benchmark: Barnes, Radix
- Cache Protocol: MESI, MOESI
- Number of CPUs: 4, 8, 16
- Topology: Mesh, Crossbar

All other parameters are identical to those given in the assignment description. No simulations used the maximum instruction limit command line flag.

Plots

The following plots aim to show how varying the number of processors impacts performance and network messages. Network messages is measured as the total messages figure given by Ruby. Performance is measured via the simulation time reported by GEM5. Normalization occurs across benchmarks. That is, it is valid to compare between benchmarks as all results from the same benchmark are normalized using the same maximum.

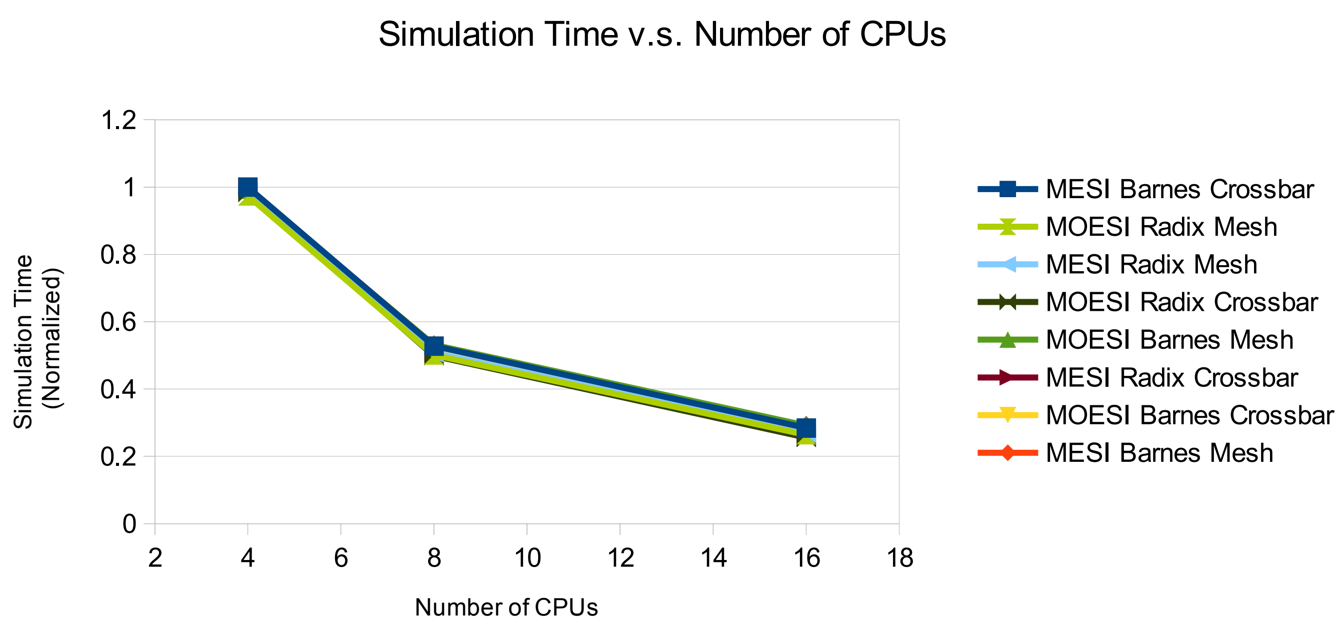

Fig. 1. Simulation Time v.s. Number of CPUs

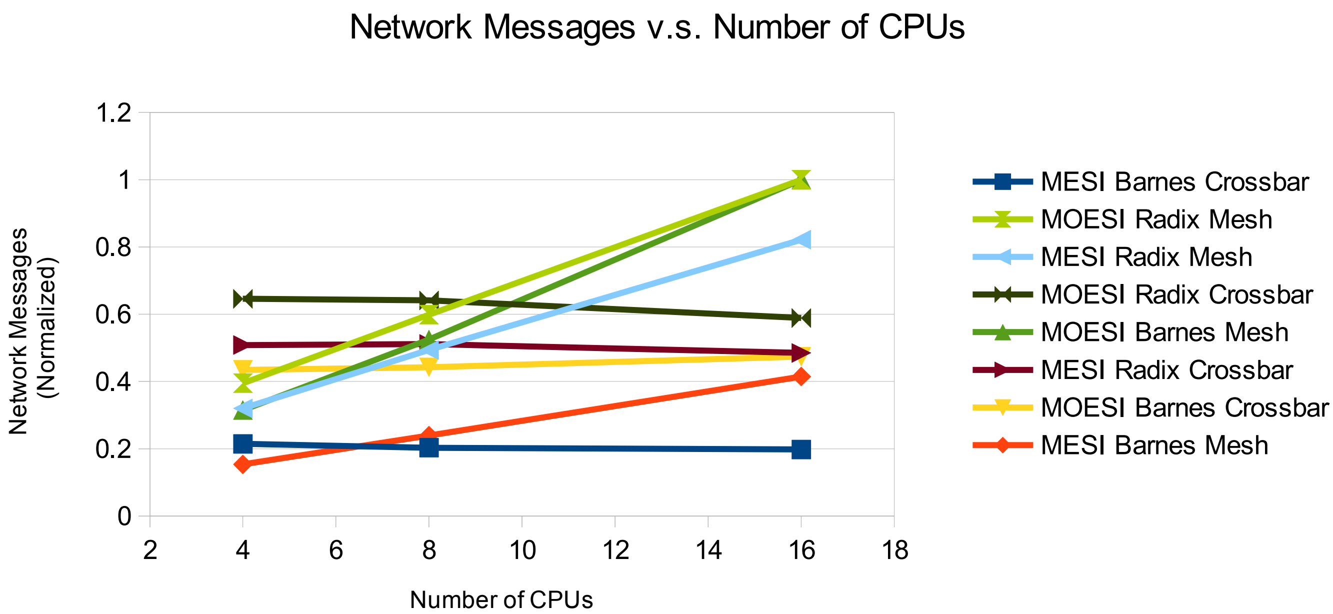

Fig. 2. Number of Network Messages v.s. Number of CPUs

Analysis

Referencing Fig. 1, it seems that the simulation time behaves similarly for all benchmarks, cache protocols, and topologies: as the number of CPUs increases the simulation time appears to exponentially decay (diminishing returns).

Referencing Fig. 2, it appears that topology is the largest contributing factor to the behavior of network traffic as the number of CPUs increases. The 'mesh' topology shows an increasing number of network messages while the 'crossbar' topology shows unchanged or slightly decreased network traffic.

When considering trade-offs it is worthwhile to note that the crossbar topology appears to be a better choice overall. This is supported by the fact that, with either topology, performance tends to increase (simulation time decrease) however it is the crossbar topology that shows decreasing network traffic as the number of CPUs increases. That is, the crossbar topology appears to scale better into multi-processor systems. This seems counter intuitive when considering the hardware implications of larger and larger crossbars (as opposed relatively well scaling mesh topologies). It is likely that the benefit of mesh topologies do not begin to show until larger (into the exascale realm) numbers of CPUs are used. One must consider these topology driven network traffic behaviors when making decisions on cache architectures for multi-processor systems.

Assuming that a 10% increase in network messages costs roughly 1% increase in energy I would choose specifically the crossbar topology due to its trend towards decreasing network traffic and either cache protocol (it is difficult to see any cache protocol specific differences in network messages or performance).

On the effect of doubling the L1 cache size (both I and D): I speculate that by having a larger L1 cache at least these two things would occur: 1) performance would increase as fewer cycles would be spent retrieving data from higher cache levels and 2) network traffic would decrease as fewer caches transactions would be needed since the L1 is larger.