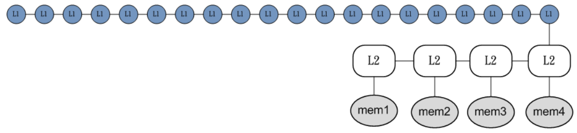
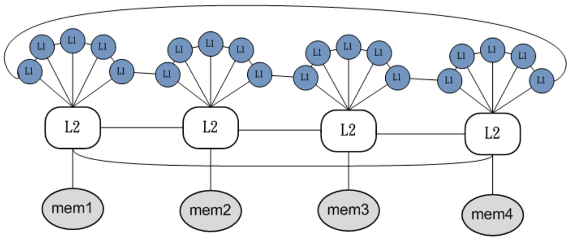
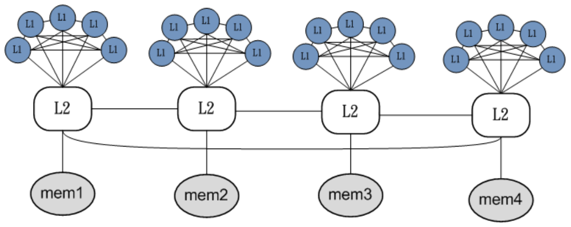
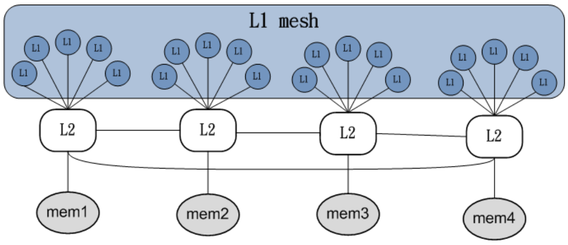
Computer Architecture Final Project Report

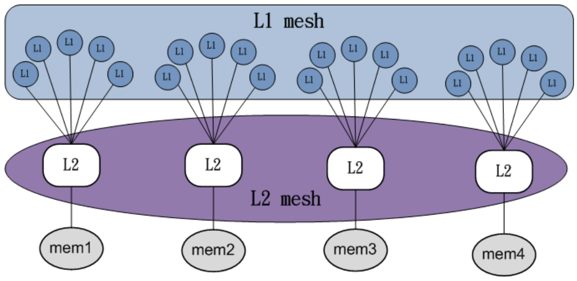
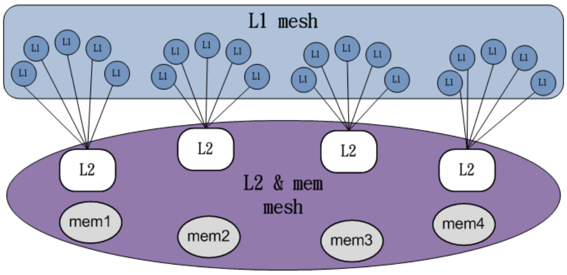
0250187 戴紹恆, 0250202 羅國強, 0250205 徐維伸

I. Hardware Configuration Overview

We first introduces various hardware configurations used in this project.

Linear Bus	Tree & Ring
	
This is the config comes with the package.It's the most cost-effective compared to other topologies. It works really slow since misses are passed through the whole network.	In order to improve our performance comparing with Linear Bus, we use tree topology that each L1 can access L2 directly and use Ring topology to increase the connectivity of each L1 tree and L2s.

L1 Four Mesh Blocks & L2 Ring	L1 Mesh & L2 Ring
	
In this config, L1 are grouped and form 4 local meshes. All nodes cooperate in the distribution of data in the network. It improves the performance comparing with Tree.	In this version, we try to let all L1 to form a big mesh.

L1 Mesh & L2 Mesh	L1 Mesh & L2+mem Mesh (future work)
	
And then we try to mesh all L2s to increase the connectivity in L2s.	We will try to use mesh topology in the connections of L2 and memory. (not finished so far)

	Advantages	Disadvantages
Bus	1) Easy to setup and maintain. 2) Low cost.	1) Total bandwidth is limited. May suffer from delay when traffic is heavy.
Ring	1) Each node has equal access to resources. 2) Each node gets to send the data when it receives an empty token. This helps to reduces chances of collision.	1) If one workstation or port goes down, the entire network gets affected. 2) Bandwidth is limited by the link.
Tree	1) If one segment is damaged, other segments are not affected. 2) Error detection and correction is easy.	1) Because of relying heavily on the main bus cable, if it breaks, whole network is crippled. 2) As more and more nodes and segments are added, the maintenance becomes difficult.
Mesh	1) Data can be transmitted from different devices simultaneously. This topology can withstand high traffic. 2) Even if one of the components fails there is always an alternative present. So data transfer doesn't get affected.	1) There are high chances of redundancy in many of the network connections. 2) Overall cost of this network is way too high as compared to other network topologies.

II. Program 1: Matrix Multiplication

Bottleneck Analysis

1. As the size of matrix grows, CPU will compute these simple but repeated calculations longer than GPUs.
2. The program has to access the values of entries serially for the computation of GPUs.

Optimization Techniques

1. Based on AMDAPP SDK sample code, CPU and GPUs working in concert enables this application to speed up. GPUs are used to compute each of product independently while CPU is used to execute the rest of the instructions.
2. The net configuration is optimized by setting up more connections between caches near different CPU in order to have more chances to access the data near the executed entries.

Experiment Results

Hardware config	SimTime (ns)	CPU cycles	GPU cycles
Linear Bus	206140992	206140571	75789
Tree & Ring	156799348	156798927	54552
L1 Four Mesh Blocks & L2 Ring	161927852	161927431	56698
L1 Mesh & L2 Ring	153070958	153070537	53527
L1 Mesh & L2 Mesh	128699497	128699071	53977

Conclusion

With L1 Mesh & L2 Ring net configuration, heterogeneous system accelerates this application the most. Because matrix multiplication needs to access the entry values serially, accessing caches near different cores using L1 mesh improves the miss rate of GPUs a lot.

III. Program 2: DCT

Bottleneck Analysis

memory barrier

1. **D**CT matrix in global memory but being read frequently

Optimization Techniques

1. Copy the DCT into local as cache in each group
2. Memory IO optimization through hardware configuration.

Experiment Results

Hardware config	**SimTime (ns)**	**CPU cycles**	**GPU cycles**
Linear Bus	379648334	379478443	928523
Tree & Ring	377844851	377844434	573712
L1 Four Mesh Blocks & L2 Ring	366963065	366962644	572065
L1 Mesh & L2 Ring	369089908	369089491	556528
L1 Mesh & L2 Mesh	366641101	366640680	553811

Conclusion

L1 mesh & L2 mesh works the best. Since the amount of memory access is large in this program. the performance improvement (in GPU) is huge. Changing network connection can largely improve performance.

IV. Program 3: Floyd Warshall

Bottleneck Analysis

1. Branching. The dynamic-programming base of this algorithm have lots of branches finding minimum. Other than that, the IO part of this algorithm is pretty parallel.

Optimization Techniques

1. The branches cannot be eliminated since the algorithm. The program can be optimized through accelerating memory operations.

Experiment Results

Hardware config	**SimTime (ns)**	**CPU cycles**	**GPU cycles**
Linear Bus	105955782	105955349	11638288
Tree & Ring	97049167	97048746	7048854
L1 Four Mesh Blocks & L2 Ring	97186106	97185674	7399737

L1 Mesh & L2 Ring	96773834	96773403	6453511
L1 Mesh & L2 Mesh	94758059	94757638	6466255

Conclusion

Similar as DCT, this program performs DP algorithm, which requires a large amount of memory IO. Plus the amount of data (adjacency map) is large, many capacity misses must appear. So changing network type can get large performance gain.

V. Program 4: Recursive Gaussian

Bottleneck Analysis

1. There are multiple kernel calls per execution, meaning that the overhead of kernel call will be doubled.
2. Barrier

Optimization Techniques

1. Since we cannot modify the main program. The kernel calls cannot be merged.
2. Mainly optimizing memory IO through hardware configuration.

Experiment Results

Hardware config	**SimTime (ns)**	**CPU cycles**	**GPU cycles**
Linear Bus	265605494	265605077	8098512
Tree & Ring	262048037	262047620	6122385
L1 Four Mesh Blocks & L2 Ring	262999259	262998831	6013310
L1 Mesh & L2 Ring	263827254	263826837	6064842
L1 Mesh & L2 Mesh	262248465	262248050	6212417

Conclusion

The performance gain of changing L2 structure is not obvious, since in this program, the amount of data is not large. Hence there is not many capacity misses in after L2. So miss time does not affect much.

VI. References

1. "Mesh Topology" [Online] Available: http://www.ianswer4u.com/2011/05/mesh-topology-advantages-and.html
2. "Ring Topology" [Online] Available: http://www.ianswer4u.com/2011/05/ring-topology-advantages-and.html
3. "Bus Topology" [Online] Available: http://www.ianswer4u.com/2011/05/bus-topology-advantages-and.html
4. "Tree Topology" [Online] Available: http://www.ianswer4u.com/2012/01/tree-topology-advantages-and.html
5. "Network topology: From Wikipedia, the free encyclopedia" [Online] Available: http://en.wikipedia.org/wiki/Network_topology