Below are the results from the analysis of the baseline scenario, with uniform delays (results obtained from previous assignment and verified again), as well as with the NUCA case. As can be seen from this, the latency will be higher for all cases (of NUCA with regard to the normal case). This is because we have the original latency of 20, but now with an added delay from the equation of 20 + 4\*(i/4). In other words, each bank access will still require 20 clock cycles to perform its respective fill/access/wb, and then the routing costs will affect the latency whenever accessing the NUCA LLC as well. For this reason, the IPC was generally lower.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Normal |  | NUCA |  |
|  | IPC | Latency | IPC | Latency |
| applu | 0.764 | 20 | 0.765 | 21.184 |
| apsi | 0.099 | 20 | 0.097 | 24.163 |
| applu~apsi total | 0.863 |  | 0.862 |  |
| art\_470 | 1.043 | 20 | 1.023 | 27.873 |
| bwaves\_06 | 0.291 | 20 | 0.275 | 28.092 |
| art\_470~bwaves06 total | 1.333 |  | 1.298 |  |
| bzip2\_source | 0.044 | 20 | 0.044 | 24.087 |
| cactus\_ADM | 0.944 | 20 | 0.939 | 23.918 |
| bzip2\_source~cactus\_ADM total | 0.987 |  | 0.983 |  |

Below is the chart comparing the average latency for the single benchmarks for Normal and NUCA cases:

We can clearly see that the normal case under these memory conditions provide lower latency for all benchmarks.  
  
Below is the IPC of the Normal case vs. the NUCA case:  
  
We can note that the IPC is very similar in all cases, but that without NUCA, there is a superior value of IPC. This is strange, as NUCA was generally superior but that developing the surrounding logic to support NUCA is sometimes not worth the effort because it could lead to more of a cost than a benefit. Perhaps the reason for the discrepancy in the value is due to the memory allocation of 8MB for the cache and that if this were different, we would see a superior performance by NUCA, or that there was some small issue with my computation.