Power Breakdown of NUCA

Vijay Daultani
Advisor: Prof. M. Balakrishnan and Ass. Prof. S.R. Sarangi

May 12, 2013

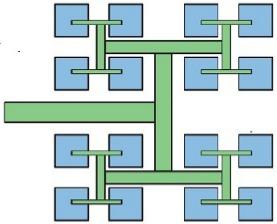
Table of contents

Goal

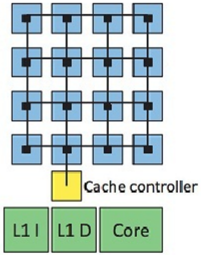
- Study Power Breakdown of NUCA cache for different programs by varying the hardware parameters.
- Insert Counters
    - No of buffer access
    - No of Link access

Cache Architecture

- **Uniform Cache Access :**
  - Delay for the cache access is set to be the worst-case delay for any block.
- **Non Uniform Cache Access :**
  - Variable access time for L2 are acceptable.
  - Instead of adopting an H-tree topology, a grid topology is employed to connect banks to the cache controller.



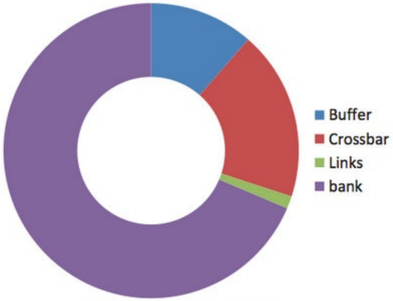
H-tree Network for UCA Architecture



NUCA L2 cache connected to a single core

Source : Balasubramonian and Jouppi and Muralimanohar. *Multi-Core Cache Hierarchies*. U.S : Morgan and Claypool, 2011

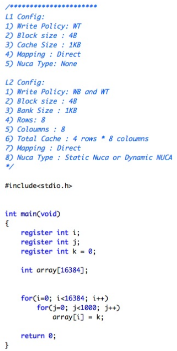
Energy Breakdown of NUCA



Energy breakdown of a NUCA

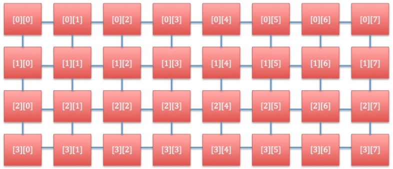
Source : Balasubramonian and Jouppi and Muralimanohar. *Multi-Core Cache Hierarchies*. U.S : Morgan and Claypool, 2011

Test Program and Cache Bank Placement



```
/**********************
L1 Config:
1) Write Policy: WT
2) Block size : 4B
3) Cache Size : 1KB
4) Mapping : Direct
5) Nuca Type: None

L2 Config:
1) Write Policy: WB and WT
2) Block size : 4B
3) Bank Size : 1KB
4) Rows: 8
5) Coloumns : 8
6) Total Cache : 4 rows * 8 coloumns
7) Mapping : Direct
8) Nuca Type : Static Nuca or Dynamic NUCA
*/

#include<stdio.h>


int main(void)
{
    register int i;
    register int j;
    register int k = 0;

    int array[16384];


    for(i=0; i<16384; i++)
        for(j=0; j<1000; j++)
            array[i] = k;

    return 0;
}
```

Test Program executed for the Experiments

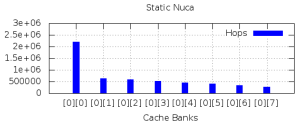


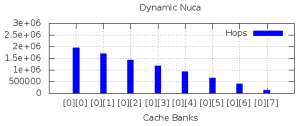
L2 Cache Bank's Placement for the Experiments.

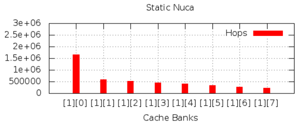
1 : Static and Dynamic NUCA

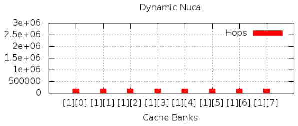
- Mapping Policies :
  - Static NUCA :
    Each memory block is uniquely mapped on a specific cache bank in the L2 cache.
  - Dynamic NUCA :
    Each memory block is not uniquely mapped on a specific cache bank in the L2 cache, rather it may present in any of the cache banks of a unique coloumn of cache banks of L2 cache.
- Demonstrate :
  - In this experiment I had tried to expose the effect of two different cache Mapping policies of NUCA architecture on the difference of hop counts of all cache banks.

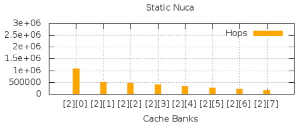
Static and Dynamic NUCA : Result for 100 access

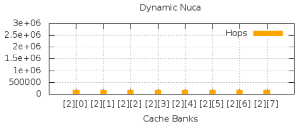


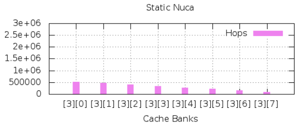


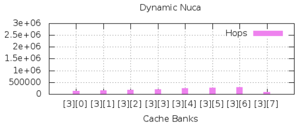




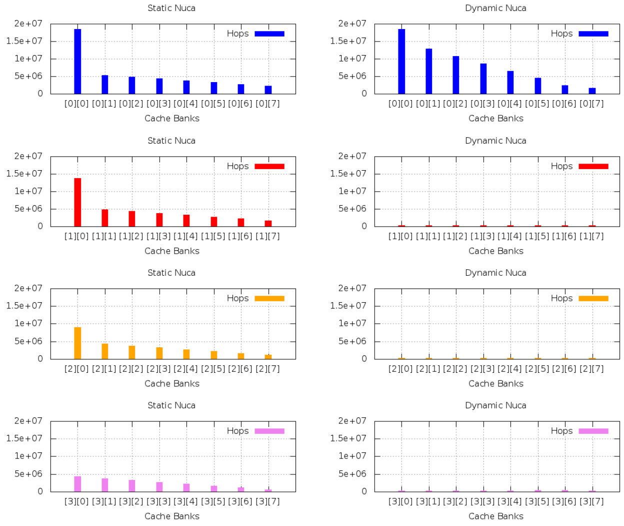








Static and Dynamic NUCA : Result for 1000 access



Static and Dynamic NUCA : Conclusion

Static NUCA

- Result shows access to each block requires data to be fetched from the unique cache bank, doesn't matter that bank is nearer or far to the cache controller. Hence may require more hop counts for a single request to be fullfilled.

Dynamic NUCA

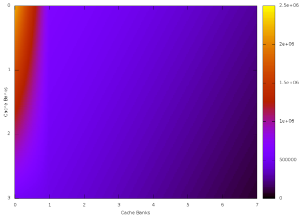
- As seen by promoting the block by seeing its hit event, towards the cache banks which are much nearer to cache controller, reduces the number of hop counts for a single request to be fullfilled.

Hence using Dynamic mapping policy will help us to reduce the time delay, after the request and before the reply of the request.

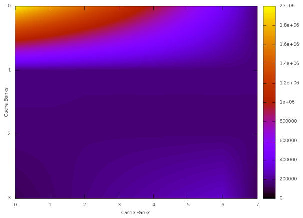
2 : Expected Power Distribution and Heat Generation

- In this experiment I tried to show that which are the areas of the NUCA cache in configurations of static and dynamic NUCA with both Writeback and Write Through policy that are likely to consume more power and hence generate more heat.

Static and Dynamic WB NUCA : Result for 100 access to each Block

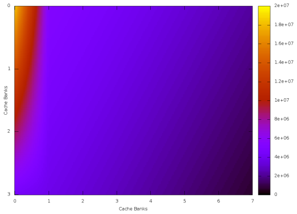


Static Write Back NUCA for 100 access for each Block.

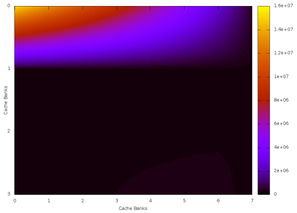


Dynamic Write Back NUCA for 100 access for each Block.

Static and Dynamic WB NUCA : Result for 1000 access to each Block



Static Write Back NUCA for 1000 access for each Block.



Dynamic Write Back NUCA for 1000 access for each Block.

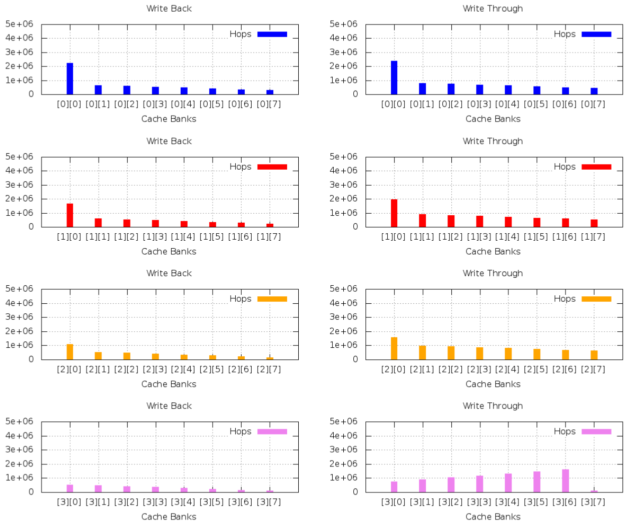
Thermal : Conclusion

- Static Write Back NUCA :
  - As was expected in case of Static Write Back NUCA all of the banks were equally accessed and thus shows the same distribution in both the graphs.
- Dynamic Write Back NUCA :
  - As was expected in case of Dynamic Write Back NUCA when the blocks get hit they are promoted to the cache banks of the same column near cache controller and thus once they reach the 1st row the request for the same data is fullfilled from 1st row itself.

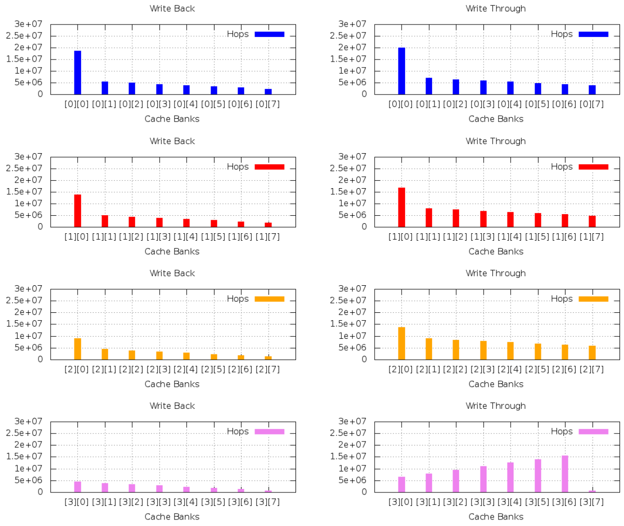
3 : WriteBack and WriteThrough

- By Defination :
  - As in comparison to WriteThrough policy, WriteBack policy requires much less number of main memory accesses because Writeback policy tries to absorb all the writes to same block in L2 cache, unless that block is evicted out from L2 cache.
- Show :
  - I tried to show the difference in hop counts of cache banks for this two different write policies.

WriteBack and WriteThrough : Result for 100 access to each Block



WriteBack and WriteThrough : Result for 1000 access to each Block



WriteBack and WriteThrough : Conclusion

As L1 cache write policy is WriteThrough each write by the processor arrives L2 cache for updation.

WriteBack Static NUCA

- Intially in case of 100 writes each write to block was absorbed and so was in the case of 1000 writes. So the block writes were absorbed till the block was finally evicted from the cache and was written to the main memory.

WriteThrough Static NUCA

- In both the cases of 100 and 1000 writes each block write updated the L2 cache copy as well as travels long towards the memory controller, which sends an updation message to the main memory. Hence increasing the number of hop counts for each cache banks.

Refernces

- Balasubramonian and Jouppi and Muralimanohar. *Multi-Core Cache Hierarchies.* Mark D. Hill: Morgan and Claypool, 2011
- A. Ho, K. Mai, and R. Balasubramonian. Non-Uniform Power Access in Large Caches with Low-swing Wires. In Proceddings of HiPC, 2009 DOI: 10.1109/HIPC.2009.5433222 93

Thankyou