International Journal of Grid Distribution Computing
Vol. 8, No.5, (2015), pp.287-302
http://dx.doi.org/10.14257/ijgdc.2015.8.5.29

# Research on the Management Strategy of the Last Level Cache Sharing Multi-Core Processor

Yuhuai Wang[1, *], Huixi Zhang[1], Yaping Sun[1] and Qihui Wang[1]

**Address correspondence to Yuhuai Wang; Email: wyiya@hotmail.com*
*[1]Qianjiang College, Hangzhou Normal University, Hangzhou 310036, China*
*Email: wyiya@hotmail.com; zhhuixi@126.com; 740756646@qq.com; wangqihui_xinyi@hotmail.com*

***Abstract***

*Effective allocation of shared resources limited is a key problem for chip multiprocessors. As the processor core growth in the scale, multi thread for the shared resource limited system competition will become more intense, the performance of the system will also be more significant. In order to alleviate this problem, a fair and effective multi thread shared resources allocation scheduling algorithm is important. In all kinds of shared resources, the largest effect on the system performance is the shared cache and DRAM system. There are essential differences between the last level cache and a cache. The goal of a cache design is to provide fast data processor which requires high access speed. However, the object of the last level cache is to save data in the chip as much as possible, and the access speed requirements are not too high, it is more subject to the plate number of available transistors. Management level cache LRU strategy and its approximate algorithm are not applicable to the large capacity last level cache for traditional. It may cause destructive interference between threads, cache thrashing of stream media program lead, which will lead to a decline in the performance of processor. This paper focuses on the analysis of some hot problems of the last level cache management in the process of the large capacity of multi-core platform sharing, and puts forward the corresponding costs less.*

***Keywords****: Chip multiprocessor, last level cache, cache partitioning, Memory access*

## 1. Introduction

In a CMP on the concurrent execution, CMP multiple cores/threads are commonly shared last level cache (LLC), DRAM controller, memory bus bandwidth, pre-fetch unit and other resources. The system of shared resources are always limited, threads need to compete for the right to the use of shared resources. Different threads of the program characteristics are different, and the thread for the shared resource capacity is usually associated with this [1，2]. If there is no scheduling mechanism, some threads may take up most or all system resources of the other threads, requesting not to service. If there Exists a scheduling strategy which can in multiple parallel threads of fair and effective allocation of resources, it can alleviate the negative effect on the shared resources, which can improve the system performance [3]. With the development of high performance microprocessor, the memory wall problem becomes one of the key factors restricting the microprocessor performance. How to solve the memory wall problem became to the processor in the design process, especially the one of the hot research problems in the design of multi core processor. The gap between the access speed data processing speed and the main memory of the processor led to the processor computing ability because of data delay and greatly restriction [4, 5]. In a multi-core processor, due to multiple processor cores also compete for resources, the memory wall problem becomes more

ISSN: 2005-4262 IJGDC

prominent, so that the design of cache and cache management strategies limits the performance of multi-core processor to a great extent.

Academia by exploring the level parallelism (ILP) and thread level parallelism (TLP) of the two different ways to improve processor performance [6]. The pipeline technique is adopted to increase the performance of microprocessors by improving instruction level parallelism which is a very effective method. Pipeline technology to an instruction is divided into several levels, each processor in a cycle of executive level, thus to execute multiple instructions in parallel in the same period, which can improve the overall performance of microprocessor. The depth of pipelining is an improved version of assembly line techniques, and this technique further refined water series and reduce the water series operation, so that the processor's frequency increase [7-10]. The data correlation program itself will not only make the pipeline design with the increasing number of complicated, but also increase the possibility of conflict between different pipelines, resulting in a growing pipeline delay. At the same time, the power consumption of the processor frequency increases sharply, which makes the heat dissipation becomes a difficult problem. Furthermore, the increase will make the line processor frequency fall, which lead to high frequency but low processor [11-13]. At the same time, the verification with the increase in the number of line becomes more and more difficult. Therefore, the performance of a processor cannot be improved desirable only by improving the processor frequency [14].

At present, the microprocessor to access the main memory latency has reached hundreds of clock cycles, and the delay in the future will continue to increase [15]. Therefore, the main memory access becomes one of the key factors that severely limit the performance of microprocessors. Practice shows that the cache can significantly reduce the negative influence brought by the storage wall. But lack of the last level cache or trigger for main memory access, hundreds of cycles lead to processor pause. Therefore, the last level cache hit rate around the whole performance microprocessor [16]. The microprocessor is also commonly used in the traditional sense of the LRU method or the approximate strategy to manage the last level cache, ignore the last level cache characteristics. However, the traditional LRU strategy and without effective use of the last level cache resources cause large number of cache misses, and eventually led to the decline in overall performance microprocessor. If design a suitable management strategy according to the last level cache characteristics, the microprocessor performance will be greatly improved [17].

This study expatiates on the current cache multi-core processors and cache management strategy. This research proposes a new method to eliminate cache management strategy of low reuse block, predicts the access interval, and validate them by multi-core simulator and multi load, and analyze the performance and overhead. A divide and conquer thread aware Cache management strategy, and validate them by multi-core simulator and multi load, and analyze the performance and overhead. This study introduces a design called low reuse block elimination and re visit interval prediction management strategy. According to the characteristic that last level cache block low reuse resources occupied longer, the strategy adopted by the historical information predict low reuse block and its priority is out of data on a cache aware last level cache. And by improving the access interval prediction technique, it can predict low reuse potential and the first elimination. Experiments show that for multi core processor, this strategy weighted speedup increased an average of 9.0% than the LRU.

## 2. The Relevant Thoery Of Cache

### *A.The design of cache hierarchy*

The design of the current mainstream high performance processor adopts multi-level cache structure, which can increase cache layer, and its capacity is increasing, but the

 

access delay is more and more big [18]. The level two cache architecture are the most common processor design process, there are very few processor with three levels cache architecture. The on-chip memory space is limited, and the cost is relatively expensive, therefore, the existing on-chip cache capacity is relatively small [19].

On one hand the speed gap between processor and memory is growing; on the other hand, the limited bandwidth will increasingly make the program performance depends on the on-chip memory hierarchy [20]. Therefore, designers will design the last level cache design large enough to save more data on chip. Level two cache architecture is widely used in design the on-chip cache. However, the level two cache architecture and management strategy faces many challenges, such as nano-control line delay which due to the difference in level two cache structure caused by the non-uniform cache access (NUCA).

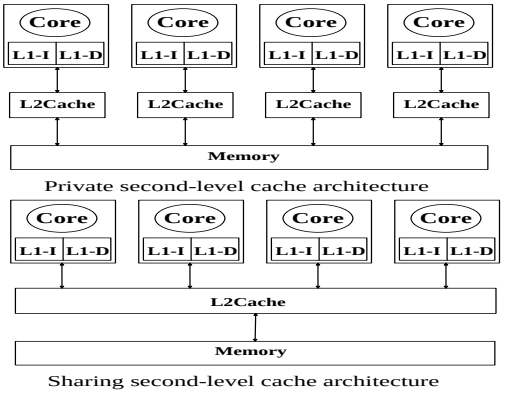


**Figure 1. The Private Level Two Cache Structure**

The design of two level caches contain two different design schemes which are private level two cache and shared level two cache. Figure 1 shows the private level two cache structure.

*B.Multiple kernel shared architecture memory*

This paper used the shared memory approach for data management. Figure 2 showed the schematic diagram of the multiple kernel shared memory structure. From the map, it can clearly see that it is connected with a bus with four modules, namely the data transmission module (ARM and FIFO), arbitration module (ARBITRATOR) module to judge, state (CHECK) as well as external storage module (SRAM).

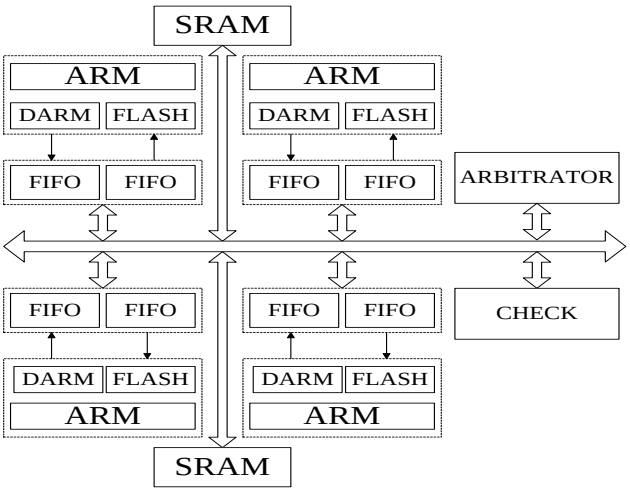

**Figure 2. Schematic Diagram of the Multiple Kernel Shared Memory Structure**

The data transmission module is composed of two parts; one part is ARM and the control of DRAM and FLASH, the other part is the FPGA internal FIFO. Using ARM as the kernel of the system, it will store their required data and information which is controlled by DRAM and FLASH. In memory management, this two parts as a class Cache to be used, stored some commonly used data to improve the efficiency of data processing. When ARM needs memory on chip for read and write operations, it should be realized by FPGA in FIFO.

*C.Cooperative cache and means of communication*

This paper used the weighted speedup and throughput fairly to measure the performance of each multi load. The weighted speedup metric is to reduce the system execution time, and the harmonic mean of fairness is the balance between performance and fairness. Application of i assuming that $IPC_i$ is a multi-channel test load of IPC, total, IPC says i keep the shared Cache when IPC.

These evaluation functions are as follows:

$$IPC_{sum} = \sum_{i=1}^{n} IPC_i \tag{1}$$

$$WeightedSpeedup = \sum_{i=1}^{n} (IPC / IPC_{total,i}) \tag{2}$$

$$Fairness_{ipc} = N / \sum_{i=1}^{n} (IPC_{total,i} / IPC_i) \tag{3}$$

The negative effect on the performance of on-chip wire delay is the main reason for private design. The utilization efficiency of the private space for Cache design is low. Thus, in the optimization of the private design basis for the design of the Cache usually focus on how to improve space and Cache utilization. Cooperative Cache is an optimization strategy for private based design of the representative; its basic structure was shown in Figure 3:

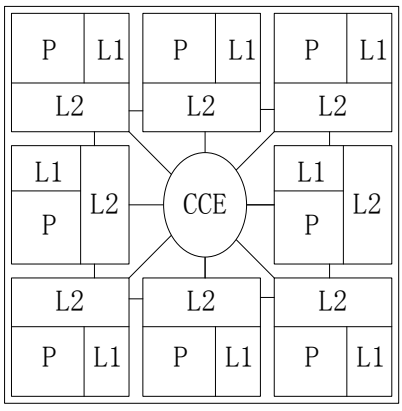


**Figure 3. Basic Structure of Cooperative Cache**

And other private design based strategy is that in the mechanism of CC, except L1 Cache, each processor core and a close from their own local L2 Cache, which can get delayed hits like private Cache. At the same time each processor local L2Cache through a set of Chinese style consistency engine (CCE). The private Cache cooperation, aggregated into a shared Cache, dirty blocks are expelled from private Cache which can be placed

 

adjacent to the processor's private L2 Cache. It can occupy a space equivalent to the shared Cache as large volume, reduce the failure rate of Cache, and reduce the off chip memory access. According to dynamic load, the number of control cooperation, can be in private and shared reasonably choosing between two Cache extreme cases which is to adapt to the dynamic behavior of the load.

*D. Cache access pattern classification*

Bumpy access mode: this mode of periodic access to a fixed length block of data, but the data block length is greater than the number of data blocks that contain Cache in size, so in this access mode Cache hit rate is low. Streaming access mode: in this access mode when a block of data is accessed, it will no longer be access; in the access mode cache hit rate is very low.

Hybrid access mode: hybrid access model is a complex of the above several access modes. In order to better explain the above different access mode, Ai said in the cache group the i cache line address, $(A_1, A_2,\ldots A_{k-1}, A_k)$, $(A_1, A_2,\ldots A_{k-1}, A_k)$ said the temporary access to cache block sequence, $(A_1, A_2,\ldots A_{k-1}, A_k)^N$ said on the sequence of continuous access to N. When k cache size, the accessed data are all in the chip, at this time, the access mode is cache friendly application access patterns, and when k > cache size, load working set is larger than the number of data blocks that contain the size of the cache, in the LRU management strategy will be accessed because the data turbulence caused 0 hit rate. When $k = \infty$, processor accessed data will never be accessed again, this time for streaming access mode. $(A_1, A_2,\ldots P_t(B_1, B_2,\ldots B_{k-1}, B_k)^M,\ldots A_{k-1}, A_k)$ said the hybrid access model, where $P_t$ represents a probability of $(B_1, B_2,\ldots B_{k-1}, B_k)^M$ access sequence, where M > = 1, l < = cache size, $k = \infty$.

## 3. The Optimization Of Cache Management Strategy

*E. CMP stage cache optimization technology classification*

The above some Cache optimization technologies such as target application, design, optimization and carried on the simple classification, as shown in Table 1:

**Table 1. Target Application, Design, Optimization Technology of Cache**

| Target Application | Design basis | Optimization objective | cache optimization technology |
|---|---|---|---|
| Single-threaded applications<br>Multi-threaded applications | Private design | To reduce the failure rate | Cooperative Cache(CC,[8])<br>Dynamic spillover receiver(DSR,[9])<br>Distributed cooperation Cache[20] |
| | Shared design | Improve the hit access speed | NUCA[1]<br>NuRAPID[2]<br>CMP-NuRAPID[3]<br>Expel block copy(VR,[4]) |
| | | Improve the hit access speed<br>To reduce the failure rate | Expel block migration(VM,[5])<br>Selective replication(ASR,[2]) |
| Multi-threaded | Shared design | To reduce | Static optimal partition[7] |

| | | |
|---|---|---|
| applications | the failure rate | Path Partition [8] |
| | | Utilization of Cache Partition(UCP, [2]) |
| | | ASP-NUCA[3] |
| | | Elastic cooperation Cache(ECC, [14]) |

It can be seen from Table 1, as private design basis for the design of the Cache due to its fast hit access speed, the optimization is focus on how to improve the Cache space utilization rate. Because sharing design has high space utilization rate, the optimization to share design is a design basis of strategy which tend to focus on two points:

1) How to reduce the influence of on-chip wire delays to improve the hit access speed;

2) How to solve a Cache space problems caused by multiple threads in CMP system sharing; these problems include how to reduce running time of total thread , and how to ensure fairness, priority support, such as QOS.

*F. Multi core interconnection structure and data elimination strategy*

When ARM writes operations, ARM writes after the data is ready and written directly for that piece of FIFO prepared in FPGA. When a packet data in FPGA, ARM are written request and header information; FPGA received the information, advanced it into the arbitration module, judging whether it satisfies the arbitration the results, if not meet, you will wait, if meet, it will enter the state of judging module; if the condition judgment module does not meet the state you would wait satisfy you to write data to the SRAM; when a packet of data was written to the ARM interrupt, notify the already finished, ARM revoke write request, this is the end of write operation. The writing process was shown in Figure 4 (a).

Read operation and write operation are very similar. Both of them contain the following steps: sending a read request first, when meet the arbitration and judgment module conditions, it will read the data into the FIFO from the SRAM, and transmits the read interrupt, after ARM received read interrupt, ARM began to read data, when the data finished reading ARM will the request for revocation. The read operation is shown in Figure 4 (b).

 

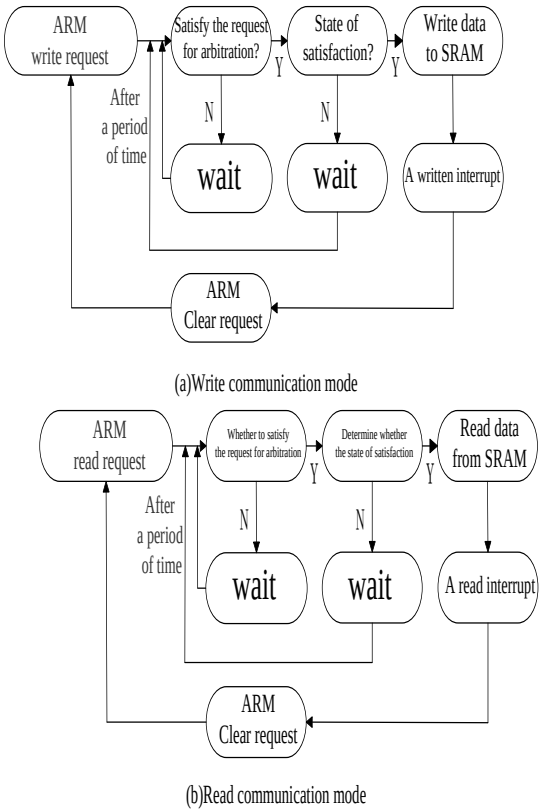


**Figure 4. The Writing and Reading Process**

The architecture of a complete NoC is composed of several NoC nodes. Each node is connected to a resource and four adjacent. Generally speaking, the node is composed of three parts including the exchanger (Switch), network interface (NI or RNI) and cyber source (Resource). Among them, the switch structure is more complex which has strong function, and it can also be called as a router (Router). Figure 5 shows the overall structure of a NoC diagram. Each switch is connected with a resource and four adjacent switches; and each resource is connected to a switch through a RNI. The resource can be a processor core, memory, a FPGA, a custom hardware module, or any other IP that can be inserted into the slot and can and network interface (intellectual property) module. In terms of current technology is relatively new, especially in the multi-processor system more complex, resource is a processor core. Switch and switch, switch and resource are connected by a pair of input and output channels. The channel is composed of two unidirectional point-to-point links.

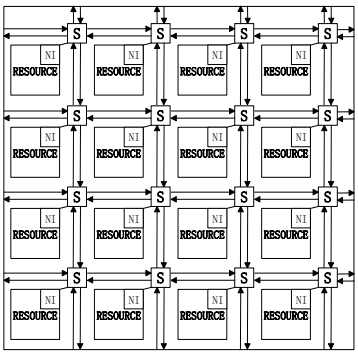


**Figure 5. The Overall Structure of a NoC Diagram**

*G. The load characteristics of multi-core platform*

Different loads in the different period of the behavior of memory access behavior are not the same; the type of load of concrete can be divided into the following three types: the efficient use of load, inefficient use of load and load saturation effect. Inefficient use of load action point and the stream media program are similar. Its working set is larger than the capacity of the Cache, even if all the resources are allocated to the load. Very few of the utility and saturated loads even for cache resources can also show good performance. The efficient use of load is between these two kinds of load, its performance can reach and resources are to a year-on-year growth trend.

Figure 6 gives some of the test cases SPEC2000 and SPEC2006 reached on different Cache allocation scheme under the lack of shared last level rate. Among them, the loss rate is in a 1MB 16 way set associative access to two levels Cache 4 Nuclear simulation platform.

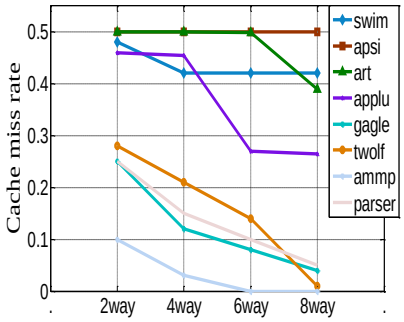


**Figure 6. The Test Cases SPEC2000 and SPEC2006 Reached On Different Cache Allocation Scheme**

From Figure 6, we can see the efficient use of load, inefficient use of load and saturation utility load differently loss behavior. As can be seen from the way, applu is an inefficient use of load, from Cache resources among its 2 Road 16 Road; its performance is not improved much remained stable in a high loss rate. Ammp is a saturated with load, even given it a little Cache resource and its absence rates remain low. While the twolf is a high performance with load, the performance increases with the growth of resources allocated to it. As can be seen from the graph, from its assigned resources to the 2 road 16 road, the loss rate in gradually decreasing, performance improving.

If several of the efficient use of a multichannel load combinations of load, or inefficient use of load and the efficient use combine together to form a multi load combination load, the interfere with how different load can be eliminated, and the problem must be solved because its working set is larger than the capacity of Cache can occur Cache bumps..

*H. The divide and conquer strategy framework*

When new data is loaded into the LLC, it will be placed in the group of set-tail, the NRU bit is set to $2^M$-2, the whether-used bit will be set to 0.

Poisson distribution, using the formula:

$$P(X = x) = \frac{\lambda^x}{x!} e^{-\lambda} \qquad (x = 0,1,...,\lambda > 0) \tag{4}$$

Type: x ----arriving customer number;

λ----A unit of time average -- "customer" reached number.

Probability density:

$$P(\mathrm{T} = t) = (\mu - \lambda) e^{-(\mu-\lambda)t} \qquad (\mathrm{t} \geq 0, \mu > \lambda) \tag{5}$$

 

T -- Business hours;

μ--The average service unit of time the number of customers

λ--Average unit in the arrival time of the number of customers

When the data was hit, there are two kinds of strategies. The one is hit priority (HP); the other is the frequency (FP). In the first case hit, hit a block NRU was set to 0; while in the frequency priority cases, hit a block NRU is based on the original 1; in these two cases, hit block is placed into the group of set-tail bi.

Figure 7 is a level two contains the Cache system structure, a structure with 2 ways set associative L1 Cache and 4 ways set associative LLC. In the beginning, there is no data of L1 Cache and LLC, the chart is access from C data. The arrows represent the direction is from the cache group of set-head to set-tail direction, which also indicated that, the LRU strategy of LRU chain, from MRU to LRU direction.

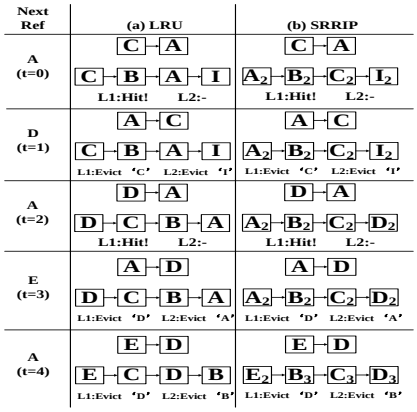


**Figure 7. A Level Two Contains the Cache System Structure**

## 4. The Cache Verification Design And Simulation Platform

Network on chip is composed of computer network development. However, compared with the computer network, the use of network on chip resource has a very high demand, because it directly affects the network on chip area and power consumption. Therefore, the design requirements of the on-chip network are to improve resource utilization as much as possible. In the routing node structure of the classic, the size of the cache for NoC resources plays a decisive role.

For this system, it needs to use the queuing model to determine the size of the cache. Each FIFO can be considered as a queue. Because the topology and routing algorithm are different, there is a great difference in the modeling method, no matter what kind of routing structure based on queuing model for modeling and calculation in the calculation of the cache size.

The theoretical basis is the queuing theory queuing model. Queuing theory, stochastic service system theory are mathematical theories to study the relationship between the "service" system, "service" and "demand", which is the operations research in probability theory, an important branch of based.

## 5. Experimental Results

*I. The performance speedup*

Figure 8 shows a DRRIP, on a single core case DIP and DELRRIP three Cache management strategy throughput speedup all data based on the traditional LRU strategy is severed as a benchmark; in the given graph of all DELRRIP are obviously beyond LRU, throughput speedup the ensemble mean value is 2.72%; at the same time, compared with DIP and DRRIP strategy, DELRRIP LLC increased by 1.76% and 1.24% respectively.

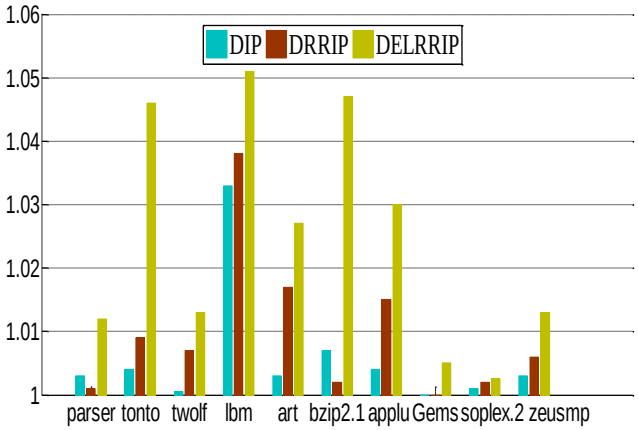


**Figure 8. DRRIP, DIP and DELRRIP Cache Management Strategy Throughput Speedup**

Figure 9 shows the TADIP, TADRRIP and TADELRRIP three cache management strategy throughput speedup, all data based on the traditional LRU strategy is severed as a benchmark; in the given graph, the 10 load TADELRRIP obviously beyond LRU, throughput speedup the ensemble mean value is 12.85%; at the same time compared with TADIP and TADRRIP strategy TADELRRIP, the average performance of LLC increased by 5.98% and 5.36% respectively,.

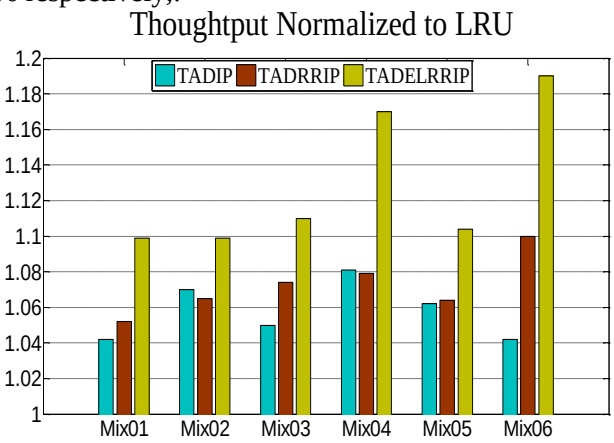


**Figure 9. TADIP, TADRRIP and TADELRRIP Cache Management Strategy Throughput Speedup**

*J. The size of the cache under Tours structure*

Each load in the share of the group can manage their own data blocks, which can eliminate the inter thread interference. Figure 10 gives the main frame of TADC strategy; the simulation platform is a 16- set associative shared two level Cache4 simulation platform. TADC will be the first 16 Cache which were divided into 4 groups. Each group contains 4 Cache block, and then the 4 groups are mapped to different application loads. From the load point of view, each of the divided group can be regarded as a 4 way set associative Cache. In this case, TADC S1 S2, S3 and S4 were mapped to load P1, P2, P3 and P4. If the P1 is a high performance with load, and P3 is an inefficient use of the load, they were in the S1 and S3 data processing block their own, so P3 cannot interfere with P1. Thus, it can eliminate the problem of interference thread.

The average waiting in the queue number or average queue length:

 

$$L_g = \frac{\lambda^2}{\mu(\mu-\lambda)} \tag{6}$$

In the system the average waiting queue length:

$$L = \frac{\lambda}{\mu-\lambda} = L_q \frac{\lambda}{\mu} \tag{7}$$

Idle rate:

$$1-\rho = 1-\frac{\lambda}{\mu} \tag{8}$$

$$P_n = (\frac{\lambda}{\mu})^n (\frac{1-\lambda}{\mu}) \tag{9}$$

TADC strategy mainly consists of three sub modules. This section first introduces divide and conquer strategy, and then describes thread behavior perception algorithm, finally will elaborates on how to divide and conquer strategy and threading behavior perception strategy together, and how to form a complete strategy to manage the shared last level cache. Described methods, examples of this section will give a TADC to better describe the strategy.

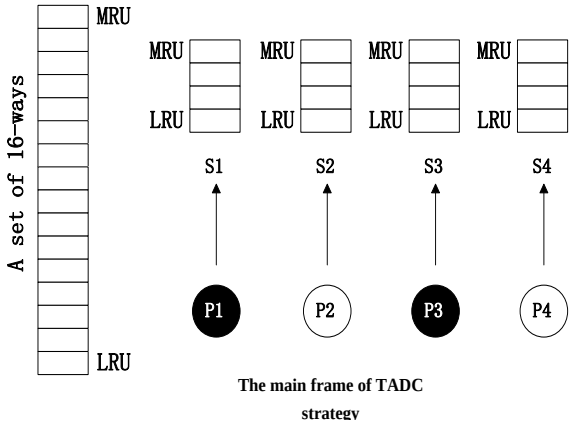


**Figure 10. The Main Frame of TADC Strategy**

Assume that each queue is isomorphic, and the injection rate of the same, can draw type:

$$\lambda^2 + \mu L_q \lambda - L_q \mu^2 = 0 \tag{10}$$

According to the formula of square root, giving invalid root, which can draw:

$$\lambda = \frac{\mu(\sqrt{L_q{}^2 + 4L_q} - L_q)}{2} \tag{11}$$

*K.Sensitivity analysis of cache capacity*

LRU and TADERRIP strategies under the average every one thousand instruction number (MPKI), where LLC capacity are 1MB, 2MB, 4MB and 8MB, and in the LLC volume change of circumstances, LLC is always 16 way set associative. As can be seen

from the graph, the TADELRRIP control strategy, LLC capacity increased from 1MB to 2MB, 2MB to 4MB and 4MB to 8MB; during these processes, MPKI increased 39.23%, 43.81% and 48.31% respectively. Therefore, TADELRRIP has good expansion capacity. TADELRRIP hardware overhead was shown in Table 2.

**Table 2. TADELRRIP Hardware Overhead**

| Overhead source | figure |
|---|---|
| The increase additional domain in each row of two-level cache | 3bits |
| The total number of rows of cache | 32K |
| Total overhead additional domain in two-level cache | 12KB |
| The saturated counter overhead | 36bits |
| The increase additional domain in each row of one- level cache | 1bit |
| The total number of rows of cache | 1KB |
| Total overhead additional domain in one-level cache | 0.5KB |
| Increased LLC area percentage of TADELRRIP | 0.58% |
| Increased area percentage of TADELRRIP in one-level cache | 0.2% |

*L. Experimental results and analysis*

*(1) Threshold analysis*

Thresholdlow and Thresholdsat values are based on experimental experience. They are 1024 and 64, respectively. We will analyze the quantitative performance based on the two values.

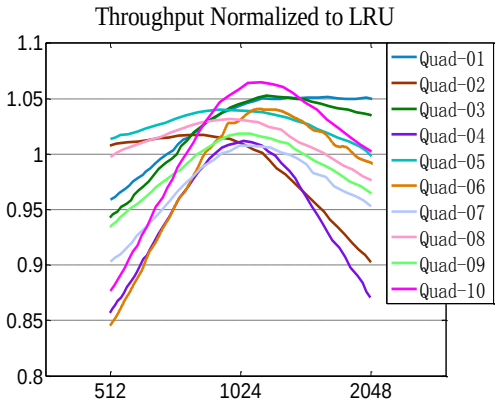


**Figure 11. The Effect of Different Threshold Low Value on the Performance of Cache**

Figure 11 shows the effect of different Thresholdlow values on the performance of Cache. It can be seen from the figure that the value of Thresholdlow changes from 512 to 2048, only the processor in 1024 near the time of peak performance, so the Thresholdlow value of 1024 is reasonable.

*(2) Throughput analysis*

Figure 12 shows the TADC, DCB and TADIP three Cache management strategy throughput speedup, all data based on the traditional LRU strategy is severed as a

 

benchmark; given by the graph of the 10 load TADC obviously beyond the LRU, for the 2 core processor, throughput speedup the ensemble mean value is 7.48%; at the same time compared with DCB and TADIP strategy, the average performance of TADC LLC increased by 4.46% and 3.87% respectively. For the 4 core processor, throughput speedup the ensemble mean value is 2.95%; at the same time compared with DCB and TADIP strategy, the average performance TADC LLC can be increased by 1.77% and 1.37% respectively.

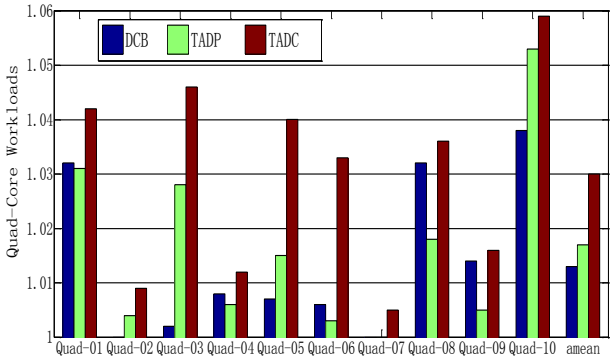


**Figure 12. The TADC, DCB and TADIP Three Cache Management Strategy Throughput Speedup**

*(3) Scalability analysis*

Figure 13 shows a LRU and TADC two kinds of strategies under the average throughput of four nuclear multiprogramming speedup, where LLC capacity are 1MB, 2MB and 4MB. And in the LLC volume change of circumstances, LLC is always 16 ways set associative. As can be seen from the graph, the TADC control strategy, LLC capacity increased from 1MB to 2MB and 2MB increased to 4MB; process throughput increased 10.10% and 20.19%, respectively. In the control strategy of TADC, LLC capacity increased from 1MB to 2MB and 2MB increased to 4MB, process throughput increased 13.97% and 24.98%, respectively. Therefore, TADC has good expansion capacity.

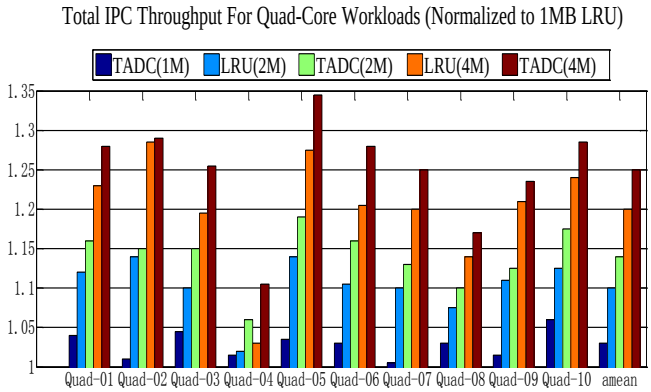


**Figure 13. LRU and TADC Two Kinds of Strategies under the Average Throughput**

## 6. Conclusions

Chip multi-processor final Cache optimization technology is the current hot spot in microprocessor research. Aiming at the problems of design of multi-processor Cache system at present, researchers have proposed many advanced methods and techniques. This paper expounds the basic ideas and features of these methods and technologies, and carried on the comparative analysis to them. In the future, all kinds of stage Cache optimization and design techniques mentioned in this paper will be an important technology which worth all of Cache system CMP and reference by designers. Because the processor computing faster than main memory access speed is two orders of magnitude, it will cause the processor due to the lack of timely access to data in main storage and long time waiting. With the development of semiconductor and multi-core technology, the memory wall problem will be more and more obvious. The cache mechanism to reduce costly for main memory access is generally used in modern microprocessor design. But lack of the last level cache access behavior will cause the main memory, the processor stall hundreds of clock cycles to obtain the required data.

## Acknowledgements

The authors wish to acknowledge the Science and Technology Department of Zhejiang Province under Grant No. 2012C31006 for their support. And our work is supported in part by Key Disciplines Foundation of electronic science and technology of Qianjiang College and the Internet of Things (IOT) engineering of Hangzhou city.

## References

[1] S. Rusu, S. Tam, H. Muljono, *etc. al*., "A 65-nm Dual-Core Multithreaded Xeon Processor with 16-MB L3 Cache", IEEE Journal of Solid-State Circuit, vol. 42, no. 1, **(2013)**, pp. 17–25.
[2] A. S. Leon, K W. Tam, Jinuk, *etc. al*., "A Power-Efficient High-Throughput 32-Thread SPARC Processor", IEEE Journal of Solid-State Circuit, vol. 42, no. 1, **(2011)**, pp. 7-16.
[3] S. B. Wijeratne, N. Siddaiah, S. K. Mathew, *etc. al*., "A 9-GHz 65-nm Intel Pentium 4 Processor Integer Execution Unit", IEEE Journal of Solid-State Circuit, vol. 42, no. 1, **(2007)**, PP. 26-36.
[4] H. Yue-li and D. Qian, "Design of architecture for multiprocessor system-on-chip (MPSOC)", Conference on High Density Microsystem Design and Packaging and Component Failure Analysis 2011, **(2006)** June, pp. 63 – 66.
[5] D. Kim, M. Kim and Sobelman, "DCOS cache embedded switch architecture for distributed shared memory multiprocessor SoCs", IEEE International Symposium on circuits and systems, **(2008)** May, pp. 21-24.
[6] Millberg, M. Nilsson, E. Thid, R. Kumar and S. Jantsch, "The Nostrum backbone-a communication protocol stack for Networks on Chip'VLSI Design", 2012, Proceedings. 17th International Conference on **(2012)**, pp. 693-696.
[7] G. –R. Andriahantenaina, "Micro-network for SoC implementation of a 32-portSPIN network", Design Automation and Test in Europe Conference and Exhibition, **(2013)**.
[8] A. Jaleel, K. Theobald, S. Steely and J. Emer, "High Performance Cache Replacement Using Re-Reference Interval Prediction (RRIP)". In ISCA **(2010)**.
[9] J. Merino, V. Puente and J. A. Gregorio, "2012. ESP-NUCA, A Low-cost Adaptive Non-Uniform Cache Architecture", The 16th IEEE International Symposium on High-Performance Computer Architecture (HPCA).
[10] T. Anderson, S. Owicki, J. Saxe and C. Thaeker, "High-speed switch scheduling for local-area networks", ACM Transactions on Computer Systems, vol. 1l, no. 4, **(1993)** November.
[11] N. McKeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches", IEEE/ACM Transactions on Networking, **(2009)** April, vol. 7, no. 2, pp. 21-23.
[12] L. M, P. Ni and K. McKinley, "A survey of wormhole routing techniques in direct networks", IEEE Trans on Computers, **(1993)** February.
[13] E. Bolotin, I. Cidon, R. Crinosar, *et.a1,* "QNoC, QoS architecture and design process for network on chip", J. Syst, Architectllre EUROMICRO, **(2004)** February, vol. 50, pp. 105-128, 141.
[14] S. Kumar, A. Jantsch, J. P. Soininen, *et. a1*, "A Netwok on Chip Architecture and Design Methodology", Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI.02), **(2012)**.

 

[15] E. Rijpkema, K. Goossens, A. Raduleseu, *et al*, "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip, Computers and Digital Techniques", IEEE Proceedings, **(2003)** September, vol. 150, no. 5, pp. 294-302.
[16] 3GPP TS 36.300, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved UTRA (E-UTRA)", **(2008)** December. V8.0.0.
[17] M. T. Hemani, A. Kumar and S. Ellervee, "Globally Asynchronous Locally Synchronous Architecture for Large High Performance ASIC", Proceedings of the IEEE International Symposium on Circuit and Systems (ISCAS). Orlando, USA, vol. 2, **(1999)** June, pp. 512-515.
[18] "International Technology Roadmap for Semiconductors (ITRS)", 2003ed, 2005ed. Semiconductor Industry Association, San Jose, CA.
[19] S. Wuytack, J. L. da Silva, F. Catthoor and G. de Jong, "Memory management for embedded network applications", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 18, no. 5, **(1999)**, pp. 533 - 544.
[20] I. Auge, F. Petrot, F. Donnet and P. Gomez, "Platform-based design from parallel C specifications", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, no. 12, **(2012)**, pp. 1811 – 1826.

## Authors

**Wang Yu-huai**, he works currently as a Lecturer of Electronic Information Engineering in Qianjiang College of Hangzhou Normal University. He has achieved the M.S. degree in mechatronic engineering from Zhejiang University of Technology in 2006. He is a PhD candidate in plastic forming. His major interest work area is Computer Application Technology and Machine Vision, Incremental Forming.

**Zhang Hui-xi**, she works currently as a Lecturer of Electronic Information Engineering in Qianjiang College of Hangzhou Normal University. She has achieved a master's degree of Electronic Circuit and System from Zhejiang University. His major interest work area is Digital Systems Development, Data Acquisition and Signal Processing.

**Sun Ya-ping**, she works currently as an electronic information engineering teacher in Qianjiang College of Hangzhou Normal University. She has achieved a master's degree of Power Electronics from Nanjing University of Aeronautics and Astronautics. Her major interest work area is Power Electronics, Power Drive and Digital Power.

**Wang Qi-hui**, he works currently as a computer teacher in Qianjiang College of Hangzhou Normal University. He has achieved the M.S. degree of Computer Science and Technology from Zhejiang University in 2002. He is a PH.D Candidate in the School of Aeronautics and Astronautics at Zhejiang University. His major interest work area is Computer Vision, Computer Graphics and Image Processing.

 