Research on the Cache Performance Optimization Technology of Multi-Core Processor Chip

Su Zhang[1, a]
[1]Department of Computer Science，Gansu Normal University for Nationalities， Hezuo Gansu ,747000, China
[a]376788368@qq.com

Keywords: Cache, Performance Optimization Technology, Multi-Core Processor Chip

Abstract. In the dual drive of process and application, multi-core structure has become the current trend of high-performance microprocessors. The competition of multi-core single-chip limited cache and bandwidth will further highlight the bottleneck of memory access. With the development of multi-target application, the performance evaluation environment of micro-architecture is facing new demands. This paper focuses on the multi-core processor on-chip cache performance optimization technology.

Introduction

With the gradual shift from deep sub-micron integrated circuits to nano-technology, high-performance microprocessor architecture is facing new challenges and innovative opportunities. According to the semiconductor industry in the world jointly developed in 2003 the international semiconductor technology roadmap and its 2004, 2005 update, the next 15 years will continue according to Moore's Law, the rapid development of integrated circuits. The continued reduction in feature size leads to an increase in integration that will enable billions of transistors to be accommodated on a single chip. Advances in semiconductor technology provide processor chip designers with more resources to implement higher-performance chips, potentially creating more complex and flexible systems on a single chip. At the same time, the target workload is constantly changing, from the early scientific computing to today's personal desktop applications, server transactions, e-business applications and a variety of embedded applications coexist. In the face of application requirements and semiconductor technology to enhance the level of the dual push in the mass integration of how to build high-performance microprocessors become an important research topic.

The Multi-Core Processor Architecture

In order to effectively rely on Moore's Law to further extend the process to support the construction of higher-performance microprocessor, you need to rely on mining and development of higher levels of parallelism design ideas and methods. The increasingly complex structure of the processor design has come to an end. Future microprocessor chips need a simple, distributed control structure, that is, more and more emphasis on the structure of the chip structure of the hierarchical, functional components of the modular and distributed, so that each feature is relatively simple, part of the internal connection as much as possible to maintain the local.

In this case, a higher degree of parallelism on-chip multi-core structure came into being. Chip-on multi-core processors (CMPs) were developed in the early 1990s by researchers at Stanford University. The idea was to integrate multiple processors on a single chip with rich transistor resources Core, parallel multi-core implementation of the way through the development of instruction-level, thread level and other levels of parallelism to improve performance.

In the private structure, Cache hit the local high rate, and the hit delay is low, but there will be a copy of data between the multi-core capacity utilization is insufficient; in the shared structure, the capacity utilization is high, but the access delay is large. This paper will study the appropriate size of the on-chip Cache level, including the optimization of the topology and the new Cache coherence

protocol to take full advantage of on-chip valuable Cache resources to improve performance. In this paper, we will use the Cache technology of capacity sharing in free nodes to make full use of the on-chip capacity while keeping the distance locality of Cache access, so as to reduce the bandwidth of off-chip access memory.

The Cache Coherence Protocol

Consistency refers to the problem of inconsistencies in multiple cache copies of the same data row in a shared storage system. When multiple processors share a unified address space, the existence of Cache will lead to the same data unit will appear in the system multi-processor copy, and the lack of centralized system, the overall global memory maintenance and management mechanisms, how to maintain between the multi-processor The consistency of the global access of the same data unit is the problem that the Cache Consistency Protocol needs to solve. Cache coherency is defined as the memory system is considered to be consistent if the access to all copies of row x in the cache satisfies the following requirements: ¹ If processor P writes X to X after completing write operations to X, Read the operation, during which no other processor on the X write operation, then return to the P write value; º in the processor Q to complete the write operation after X by the processor P X read operation, and the write operation And the read operation is completely separate during which no other processor writes to X returns the value written by Q • Multiple different processors write to the same cache line in the same order for all processors visible.

From a hardware implementation point of view, Cache coherence protocol requires the relevant processor of the system to access the data unit in a cooperative way, realizing two functions, how to locate and obtain valid data and how to propagate the new value. Cache coherence protocol can be divided into write invalid and write update according to adopting and writing and propagating the tactics. Writes are invalidated by invalidating other copies of the cache to complete the notification of data writes. Other processors need to re-access to get the latest data; write updates directly to the new value will be written to all other copies of the spread. As the write update protocol requires a high degree of data consistency at any time, requiring a lot of overhead to update all the copies, the cost is high, so most of the parallel shared storage systems are designed to write invalid agreement.

There are two main implementations of cache coherency, depending on how they are notified when they collaborate:

(1) Monitoring protocol. The listening-based conformance protocol uses a broadcast mechanism that broadcasts request information to all processors when data that needs to be read is not in the local cache or needs to communicate consistently with other processors. Each processor in the system needs to participate in any consistent request, the processor receiving the request information according to their own Cache status, return to the requester corresponding data or response to control information.

The listening-based coherency protocol is simple and effective, requiring each processor to listen to memory accesses from other processors, usually in an SMP system, where the shared bus is an efficient broadcast medium and can be used to request Return of send and reply. Broadcast mechanism for communication bandwidth requirements, and the bus can provide the bandwidth is limited, can be achieved when the sub-parity address of the multi-bus structure to improve efficiency. As the bus is an exclusive resource, its scalability is limited, so the listening protocol is usually used in a small number of nodes or small and medium-sized servers in the field of work.

(2) Directory Agreement. The directory-based conformance protocol is usually applied to large-scale distributed shared storage systems based on CC.NUMA architecture. Unlike the listening protocol, the maintenance of the directory protocol consistency state is used. Approach is consistency maintenance activities of the participants only include a copy of the data processor. The idea is to maintain a directory entry for each memory row, to record the processor number that owns the copy of the memory row and the corresponding state and control information, and each memory row has a corresponding host according to the mapping and distribution of addresses, Home node location. When the processor needs to cooperate with the rest of the processor and communication to complete the access to the Cache data, the request sent to the host node first, according to the

contents of the home node content in the directory this time. The processor that needs to be notified is then sent the appropriate request to these associated processors.

Compared with the listening protocol, directory protocol reduces the bandwidth of the broadcast mechanism to avoid interference to the normal operation of the processor, has good scalability, but the multicast mechanism needs to visit the host node, which will increase access Of the delay, in addition, the storage directory entry also requires additional hardware overhead. For the directory of the organization and management can be divided into bit vector, limited pointer, and so on. Bit vector mode in the directory for each processor in the system to assign a fixed status flag, the required capacity and the number of processors N and shared memory space M product is proportional to the commonly used in small and medium-scale distributed sharing Storage system. When the system node number and content capacity is large, the bit vector mode requires excessive hardware resources, and in most cases, the storage line is only a limited number of processors to share, so you can use a limited number of pointers to hold The processor of these variables reduces the hardware overhead to the logarithm of N. Due to the limited number of pointers, when the pointer overflow situation, can be broadcast, software replacement, etc. for processing.

The Multi-Core Storage Subsystem Design

Multi-core storage subsystems need to balance the design decisions in three areas: storage hierarchy design, interconnect topology, and Cache coherence protocol. These three aspects interact with each other, and the storage hierarchy design determines the interconnect topology and cache coherence protocol to be used. The interconnection topology restricts the storage hierarchy and cache coherence protocol that can be implemented.

On the two Cache unified addressing strategy, on-chip processor core can be directly on all secondary Cache access; and private structure of the CMP system is similar to the SMP, each processor core has its own independent two Cache, this part of the secondary cache with the processor core similar to a cache, only serve the processor core of the node cannot be directly access to other processor cores. As a result, the on-chip cache resources in the shared structure have a more balanced and full utilization, while the private structure has the advantage of access latency. Due to the abundance of CMP target applications, different programs will exhibit significantly different access characteristics, and even different stages of the same program will also be very different performance. Faced with the diversity of application behavior characteristics, shared or private secondary cache structure is difficult to provide a good range of different applications support.

Multi-core shared storage hierarchy will lead to data consistency issues, the use of what Cache consistency will have a significant impact on CMP structure and performance. Based on the consistency protocol, it is necessary to monitor the cache status of all nodes across the whole chip. It is difficult to overcome the influence of line delay and face great difficulty in process implementation. Therefore, the actual multi-core chips adopt directory-based Cache protocol. Directory protocols have different implementations for shared and private structures. In the shared cache structure, the consistency relation is maintained in the first-level Cache, and the directory is set in the second-level cache, which avoids competition for the first-level Cache port on the main water line. The consistency of the private Cache structure is maintained in the second- Can not be like a shared structure that will set the directory in the secondary cache, the solution can be achieved by copying the secondary cache Tag to achieve directory (copy TAG is the on-chip Cache directory), because the secondary Cache capacity is usually much larger than a Cache , So in the private structure by copying the two Cache Tag to achieve the directory will increase the cost of the system hardware resources.

The Cache Optimization Technology of Multi-Core Processor Chip

Tang proposed the first centralized catalog scheme for cache coherency control. The main idea is to use a centralized catalog to record all the cache conditions, including the current information and

the status of all cache lines. The centralized directory is only suitable for small-scale multiprocessor systems, such as MP860 hierarchical parallel supercomputers developed by Tsinghua University.

The Tang method allows an unmodified data block to have copies in more than one Cache, but a modified block can only exist in one Cache. Each piece of data in the Cache has a "modified" bit. The entire "modified" bits of each cache are recorded in a centralized table of contents in main memory. When the Cache write hit, you need to check the Cache in the data block "modify" bit. If the block has been modified, it is immediately written and no longer need to check the centralized catalog table. If the block has not been modified, then the Cache will notify the centralized table of contents to invalidate the copy of the block in the other cache. When a write miss occurs, the centralized catalog table is searched. If another block of data in the cache has been modified, this block will be written back to the main memory and clear the block from the Cache, then the main memory to the block provided to the request Cache. If there are other blocks in the cache that have not been modified, the copies are made invalid and then the block is provided by the main memory to the request cache. In the above-mentioned operation, the content of the centralized catalog table is changed according to the state of the data blocks in each Cache. The Tang method copies each of the Cache's own table of contents into a centralized catalog table of main memory. To find which cache has a specified copy of the data block, you must search the entire directory table, lookup overhead is too large, In addition, if several processors simultaneously to find the table of contents, there will be contention.

Censiert and Feautrier proposed a distributed catalog scheme (Censier method for short) and adopted a consistency strategy similar to the Tang method, but the organization of the table of contents is different. The catalog table of the Censier method is constructed on the basis of the main memory data block. The number of bits in the vector is equal to the number of Cache in the system. Each bit corresponds to a cache indicating whether there is a copy of the data block in the corresponding cache. Such a directory table mechanism can speed up the search. Cache as long as the pre-visit the data block address can be directly found in the main memory of the corresponding information, do not have to search the entire directory table.

Compared with the listening protocol, the system based on this directory protocol has better scalability. The directory-based protocol is applied to the disordered network. By searching the directory, the data request is forwarded only to the processor core which contains valid replicas, which greatly reduces the amount of messages between the processor cores and reduces the communication bandwidth requirement for the on-chip network. On the other hand, the cache-to-cache access miss request has to go through three jumps and the delay is long.

Conclusion

As the speed gap between processor and DRAM will become increasingly large, the competition of multi-core single-chip limited cache and bandwidth will further highlight the bottleneck of memory access. The storage system is one of the most critical factors to determine multi-core processor performance. At the same time, the research on diversity and application of multi-core structure also puts forward more requirements for the simulation evaluation environment based on the architecture research institute.

References

[1] Huifang Zhou: Journal of Information, Vol. 6 (2014) No 53, p.25-26

[2] Hongli Zhang: Computer Education, Vol. 12 (2015) No 27, p.74-76

[3] Qin Guo: Computer and Network, Vol. 1 (2012) No 33, p.11-14

[4] Jieming Liu: Guangxi Normal University, Vol. 3 (2011) No33, p.121-124

[5] Hongli Zhang: Computer Architecture, Vol. 12 (2015) No 27, p.74-76