L3/L4 Multiple Level Cache concept using ADS

Hironao Takahashi[1,2], Hafiz Farooq Ahmad[2,3], Kinji Mori[1]
[1]*Department of Computer Science, Tokyo Institute of Technology*
2-12-1 Ookayama Meguro, Tokyo, 152-8522, Japan
[2]*DTS Inc, Kyoudou Building 9F, 2-18-7 Higashiueno, Taitou-ku, Tokyo 110-0015 JAPAN*
[3]*School of Electrical Engineering and Computer Science (Formerly NIIT)*
National University of Sciences and Technology, H-12, Islamabad, Pakistan
E-mail: takahashi@cs.titech.ac.jp, mori@cs.titech.ac.jp, drfarooq@niit.edu.pk

Abstract

***Abstract*—Demand of social activity of mankind require more timeliness communication with large amount of data transfer. To meet these demands, traditional computer system architecture for IT services is weak for communication ability. The one of reason is increasing disparity between CPU speed and storage system. These systems require novel techniques to achieve high I/Os for computing systems with high assurance properties, such as timeliness, fault tolerance and online expansion. Proposed L3/L4 cache concept is balanced memory architecture that is for high I/O (Input Output) intensive Autonomous Decentralized System using Data Transmission System memory concept in DF (Data Field) to achieve timeliness in information systems to solve today's High I/O intensive application market demand. The proposed architecture also implements two new memory caches in the system, namely L_3 that works at local memory level and L_4, on top of HDD. Moreover, The remote storage also good candidate of L4 cache devise. This system design of ADS improves timeliness communication response manner. Therefore, the proposed system architecture solves previous issues and it achieves more flexible of program, data files with real time manner with system run continuously. The I/O performance evaluation shows significant enhancement over the traditional computing and ADS systems and describes potential application and services using this concept in ADS.**

Keywords- DTS (Data Transmission System) cache, Timeliness, Write back cache, Block I/O device, Locality of reference, I/OPS, Bonnie Bench mark program.

Motivation

Today's mankind life is established by world-width human communication between country to country, industry to industry and so on. Economy also requires much more fluently communication with timeliness. Now that 21-century are high demand of communication era. Under these circumstances, Information Technologies need more powerful communication technology with high level response time and large a mount of data distribution. Since 1977, Autonomous Decentralized System (ADS) is utilizing many application and situation in social environment. Once of ADS feature is multi level assurance by same hardware node. This concept brings high level data integrality and service continuity.

The emerging applications require continuous operation, non-stop service system and timeliness to achieve high assurance to meet Service Level Agreement (SLA). SLA is the explicit fulfillment of the requirement of the Quality of Service (QoS), such as reliability and timeliness. SLA requirement in the emerging applications on the Internet needs ADS (Autonomous Decentralized System) characteristics for information system. A major challenge in the realization of QoS in the Web services is the requirement of high I/O transactions for information systems. ADS concept is high assurance computer system architecture proposed in 1977 to meet the requirement of the industry, especially control systems. Theoretical foundations of ADS rely on the principles of autonomous controllability and autonomous coordinatability. These two properties assure timeliness, online expansion, fault tolerance and online maintenance of the system. But, traditional Autonomous Decentralized System does not perform high I/O intensive services for information systems on the Internet. Generally, CPU utilization ratio on the server is 10 to 15%. Therefore, people try to use CPU more effectively by Virtual OS with multiple application software on the same computer hardware. But these services place high I/O intensive requests and transactions on the server. I/O intensive applications with large, sustained workloads of various I/O sizes have special management challenges. High I/O environments are commonly thought of as a large SAN issue, however small and medium-sized storage environments can be I/O intensive, if there is a large amount of active workload and I/O processing. An environment with a relatively small number of servers and storage, for example, an OLTP, database server or vertical application for CRM or ERP can be I/O intensive. I/O intensive applications may have large, medium or small amounts of storage, yet have active workloads with either a large number of I/O requests of various sizes. For large I/O sizes, huge amount of data is moved and enormous bandwidth is utilized, while smaller I/O request consumes less bandwidth when processed. Therefore an I/O intensive application system have to be designed to support many transactions, support a large amount of bandwidth, or both. Moore's law gives rise to extremely high capacity circuits, such as large storage and powerful processors. Overall performance of CPU has been increasing 50% /year, while for storage it has been merely increasing 7% / year. The access time of a disk drive is many orders of magnitude slower than CPU cycle time. The increasing disparity between CPU speed and storage system leads toward performance degradation of the computing systems.

There have been a number of efforts to improve I/Os performance however these are substantially different from our work. In the author proposes Unified Buffer Cache (UBC), the focus is to unify the file system and virtual memory caches of file data to improve I/O transactions. However it is an unmanaged one level cache that is very much different from the DTS cache though the objective is to improve I/Os. Similarly, provide solution for high I/Os, based on solid-state disk, and is altogether different from DTS layered cache. The concept is to use non-mechanical devices to achieve high I/Os. Moreover, this solution is highly expensive compared to the DTS technologies.

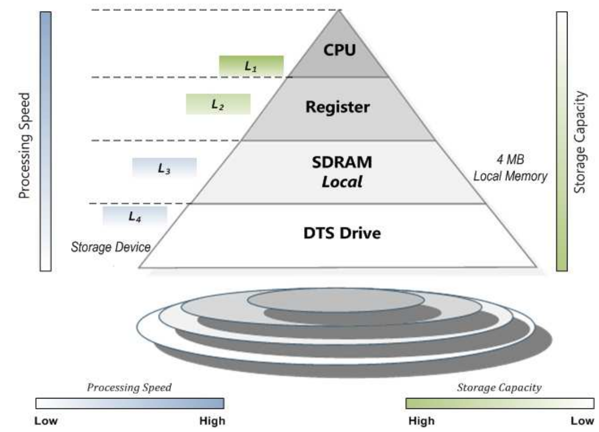
To achieve timeliness communication with high assurance demand, this paper proposes data memory layered architecture, Data Trans Mission System (DTS) on each node. This proposing concept composed two parts. One is represent t block devise as cache and other is target block devise that was mounted from CPU. At present Microprocessor has L1 cache and L2 cache memory inside of CPU. Some one has triple cache level until L3 cache. But our proposal is outside of Microprocessor and it in located system memory by DRAM and system disk as target block devise.

In this paper firstly a novel layered cache architecture consisting of 4 levels cache hierarchy has been proposed inline with memory hierarchy. Secondly ADS based ADDTS system has been proposed to achieve online expansion in the storage system on the Internet. The proposed system enhances I/O performance to achieve timeliness, and optimize the energy usage in storage systems by increasing CPU utilization and reducing frequent HDD access respectively. The rest of the paper is organized as follows. Section 1 is Proposed Balanced Memory Architecture Section 2 describes the DTS subsystem architecture. Section 3 shows L3 /L4 cache layer benefits and Section 4 gives the system design and details. Section 5 performance evaluation using benchmarking tools and network evaluation by bench mark test program. Section 6 conclusions and the future work.

1. Proposed Balanced Memory Architecture

The Concept of Architecture

Storage systems need to provide high I/O transactions in addition to their capacity and availability requirements. Generally large and medium size storage systems consist of disk arrays and associated caches to improve the I/O performance. Multi level cache hierarchy requirement for computing systems has been recognized as early as 1989. Caches at various layers of memory hierarchy in computing systems are more fast but very small compared to the adjacent lower level of storage device. This paper proposes a novel concept of layered cache based system (Figure 1). This is Data Transmission System (DTS) concept based on hierarchical layered memory architecture for information systems. The memory layers consist of hierarchical caches starting from local memory to hard disk (Figure 1 a). Proposed concept consists of 4 levels cache (L_1, L_2, L_3, L_4,) to store important data for read and write on various levels in close proximity to the CPU. The key requirement for the size of the cache at lower level L_{j-1} is that it should be smaller than the size of the cache at higher level L_j to maintain all the data of lower level cache layer at higher cache layer. This requirement fulfills the inclusion property dictating that the contents of lower level cache are subset of higher-level cache unlike the non-inclusion where this property does not hold. This property gives rise to balanced layered memory architecture as shown in Figure 1(a).



(a) Balanced computing system memory hierarchy

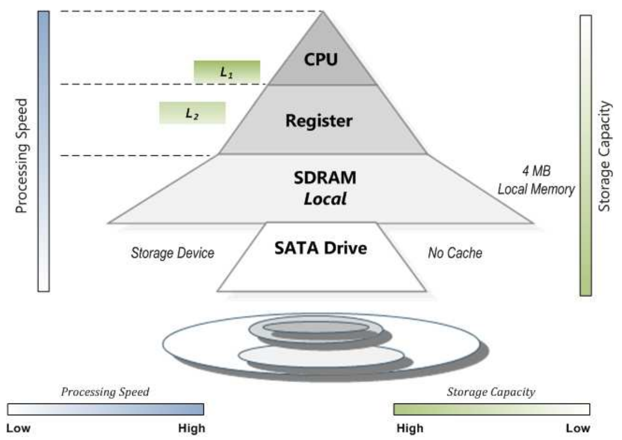


Figure 1:(b) Unbalanced computing system memory hierarchy

Figure 1(a) is DTS based abstract system architecture for high I/O intensive information system to achieve timeliness. The local memory connected to CPU via DMA is converted into block I/O cache device, namely DTS cache by dividing it into two areas. This DTS cache concept is very unique and it brings excellent potential in I/O intensive application services. The L_4 cache is placed in the specially designed hybrid HDD using DTS memory concept. DTS L_4 cache based hybrid HDD has 1GByte SDRAM as cache with HDD as target drive. DTS L_4 cache based hybrid HDD is managed by memory layer control device called DTS chip inside the drive. The cache algorithm is 100% write back policy. To achieve reliability in this context, the drive uses an uninterruptible power supply inside HDD. The DTS intelligent function loads the most frequently used block I/Os and maintain these on the cache. One example is quick boot, where a system boots up using this HDD and it pre-reads OS boot up sectors and application programs at the previous boot up process. Therefore DTS L_4 cache based hybrid HDD improves boot up time for any Operating System (OS) such as Windows XP, Windows2003, Linux and UNIX. It also shows significant performance in computer memory consuming environments such as virtual OS software as VMware and XEN. Eventually the I/O performance will be 10 to 100 times faster than traditional HDD depending on the size of data.

The balanced memory architecture facilitates efficient resources utilization while unbalanced memory hierarchy creates performance bottleneck (Figure 1b). The proposed system concept provides high I/Os as explained in the Section 2.2.

2. Subsystem Architecture

The proposed subsystem architecture is realized by creating two new caches L_3 and L_4 as $L_1$and L_2 are already part of the computing system design. Therefore, we focus on how to design and implement L3 and L4 caches. In our proposal L3 cache is realized by converting local memory into block I/O device. An appropriate cache policy is applied to manage data on this newly created cache. Write back cache policy is the most appropriate choice for transferring data on DTS cache and subsequently storing it to the HDD.

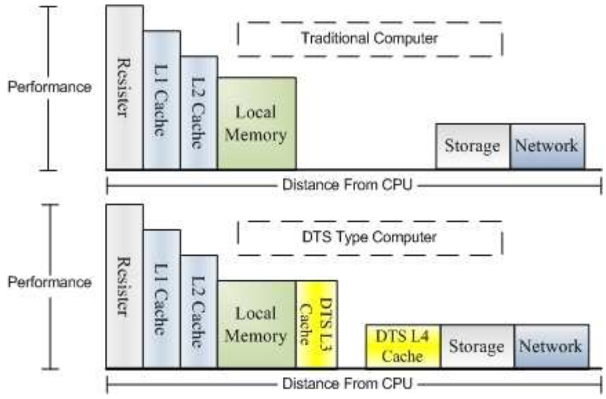


Figure 2: DTS Computing System vs Traditional Computing System architecture

Two-way I/O transactions i.e. write and read are performed on local memory in DTS type computing system. Therefore, DTS is the technology for data transmission management, which has intelligent caching function. DTS cache technology is independent of the physical device characteristics and therefore it can use any block device as DTS cache. However, it represents slow speed device at low-level layer (Figure. 2). For instance, if user needs more high I/O performance, then select the most powerful block I/O device as cache. The local memory comprising of SDRAM is the most appropriate choice for this purpose. DTS technology manages to keep the most frequently used I/O block data in the DTS cache. It proposes efficient search technique to quickly search the DTS cache. This further enhances the search time within the DTS cache, called random index search. It helps to find the required block quickly compared to the conventional search in the simple cache. A technique has been proposed in to reach quickly the desired location of the DTS cache called jumping algorithm. It also enhances network transfer speed through DTS cache utilization.

The proposed architecture has DTS based memory in each subsystem node and caches the block data from HDD device. The user program on the subsystem uses 4GB in case of 32bit OS. These user programs are located in memory or HDD depending on the situation of memory in the subsystem. Therefore, there is no guarantee that the cache stores all the data from HDD. Proposed memory architecture maintains I/O data for each user if data is required frequently.

In this section though the focus is on the DTS cache in a single subsystem, but in the next section, we present some details how this cache can be used to create distributed DTS cache system using ADS principles.

DTS based cache memory configuration for high I/O intensive system architecture is shown as Figure 3, where subsystems communicate through logical DF inspired by ADS. In the context of ADS, each subsystem broadcasts data with Content Code (CC) to every other subsystem, and this memory area i.e. DTS cache at level L3, and level L_4, is used as common I/O area. This DTS cache is reserved as a common area for all I/Os in the DF in ADS. The target drive is internal HDD or remote storage connected via high-speed network. The protocol between subsystems to remote storage is fast data transmission protocol such as iSCSI.

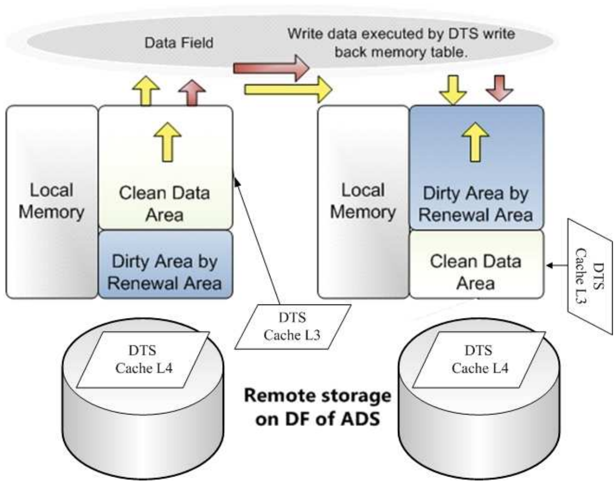


Figure 3: DTS basis cache memory configuration for high I/O intensive system architecture using Data Field of ADS.

This system architecture brings autonomous storage expansion as per the system requirements for the application. One of the subsystems broadcast write event result on DF, the second subsystem receives it and executes this event with CC. Each subsystem has write-back policy and it does not need to wait the write operation to secondary storage. Consequently, each subsystem carries out the execution of the write operation in the local memory, such as SDRAM as cache level L_3, and cache L_4 on top of DTS based hybrid memory HDD. The read/write transactions response time is enhanced due to both of these caches acting as virtual hard disk and also due to write back policy of the cache. If there is no DTS cache on the subsystem, renewal of data and reading data cannot be maintained on local memory area because it is unified memory space defined by OS policy.

3. L3 /L4 cache layer benefits.

Proposing L3/L4 cache design concept brings many benefits for high I/O demand and real time application under ADS. The first of all, L3 cache based Data Field in ADS has following benefits.

1) Timeliness data communication using DTS based memory layer management policy.
2) L3 cache memory are mounted L4 cache based block storage devise remotely. Therefore, each node is same hardware configuration with minimum storage capacity and it doesn't own programs and related data files.
3) Once received from SP information or notice such as emergency information and large data files, L3 based concept doesn't need any actual large data file transfer event.
4) Eventually, each node obtains request command and emergency notice from SP and obtain these actual large files from remote storage which was mounted by each node by iSCSI protocols.
5) Data update and write event only SP privilege event. Other nodes are receiving update status from SP and upload from remote storage by each connection.
6) Finally, the all of event L3 cache Data Field is higher speed data transfer and timeliness ADS network environment without any amount of data limitation.

Also L4 cache layer has flexibilities are,
1) There are individual block storage devises.
2) Each of them is no attribution of the group and anybody can access this group of storages.
3) When SP update the files or select and indicate the files and/or program inside remote storage, SP only notice this index information and location of files.
4) Therefore, each node mount L4 cache based storage and obtain large amount of data files.
5) L4 cache layer data field provides only data of contents.
6) L4 cache layer also provides application program for each node. This process brings non-ownership policy of application program. Then, it is easy to alternate new service node because the all of application easy to transfer to new node.
7) It also easy to enable online expansion and online program update as well.

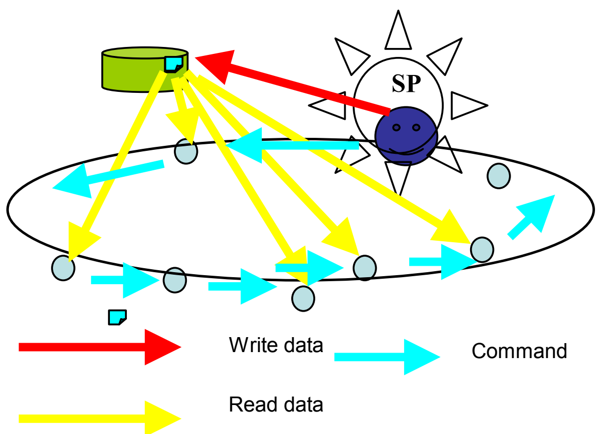


Figure 4: L3 cache based Data Field architecture

Potential application and services using L3/L4 cache concept on ADS are following.
1) Emergency and contingency information system.
2) Real time assurance system.
3) Factory Automation on demand production control system.
4) High I/O transaction information system.
5) Large multi media movie data distribution
6) High demand Web service application with high resolution of data contents.
7) Large scale data ware house

Thus, proposing concept bring more capable of data expansion and timeliness communication response.

4. Detailed System Design

DTS cache technology proposes to manage three main functions to realize the proposed system architecture. Firstly it converts local memory into I/O block device called DTS cache, and mounts this on target device, such as HDD. Secondly, the cache management policy is selected as per the application requirement, and lastly, an efficient technique is proposed to place and access the data in the DTS cache.

DTS cache concept is to use block device as a cache. DTS cache technology is independent of the physical device characteristics and therefore it can use any block device as DTS cache. However, it represents slow speed device at low-level layer (Figure. 1). For instance, if user requires low latency, means more high I/O performance, then there is need to select the most powerful block I/O device as cache. The local memory comprising SDRAM is the most appropriate choice as a cache. In this paper, local memory is transformed into a block I/O device to be used as a DTS cache. Consequently it performs high I/O transactions and significantly improves response by utilizing bandwidth equal to the local memory. The DTS cache management software detects the block address, which has been requested, and keeps this data on the DTS cache. In this way, most of the requests are fulfilled from the DTS cache. If the requested block data is not available on the DTS cache, then it is accessed from target storage device, such as HDD. The target block I/O device may be HDD in the same system or remote block I/O device attached via network. Figure 6 shows DTS block I/O data process image between cache device and target drive (secondary storage). The process of events is as follows: