



US007337240B2

(12) **United States Patent**
Bayer et al.

(10) **Patent No.:** **US 7,337,240 B2**

(45) **Date of Patent:** **Feb. 26, 2008**

(54) **VIRTUALIZATION OF I/O ADAPTER RESOURCES**

(56) **References Cited**

U.S. PATENT DOCUMENTS

(75) Inventors: **Gerd Konrad Bayer**, Stuttgart (DE);
Wolfgang Eckert, Altdorf (DE);
Markus Michael Helms, Boeblingen (DE);
Juergen Maergner, Sindelfingen (DE);
Christoph Raisch, Gerlingen (DE);
Thomas Schlipf, Holzgerlingen (DE);
Klaus Theurich, Gechingen (DE)

5,659,690 A * 8/1997 Stuber et al. 710/307
5,664,150 A * 9/1997 Isaac et al. 711/143
5,671,355 A 9/1997 Collins et al.
6,112,252 A * 8/2000 Hausman et al. 709/250
6,564,271 B2 * 5/2003 Micalizzi et al. 710/39
6,594,712 B1 * 7/2003 Pettey et al. 710/22
6,744,765 B1 * 6/2004 Dearth et al. 370/394
6,801,927 B1 * 10/2004 Smith et al. 709/202

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(Continued)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 597 days.

FOREIGN PATENT DOCUMENTS

(21) Appl. No.: **09/683,275**

EP 0772131 A 5/1997

(22) Filed: **Dec. 6, 2001**

(Continued)

(65) **Prior Publication Data**

US 2002/0099879 A1 Jul. 25, 2002

OTHER PUBLICATIONS

Microsoft Computer Dictionary Fifth Edition, 2002, p. 81.*

(30) **Foreign Application Priority Data**

Dec. 15, 2000 (EP) 00127579

(Continued)

(51) **Int. Cl.**

G06F 15/16 (2006.01)
G06F 15/167 (2006.01)
G06F 3/00 (2006.01)
G06F 13/00 (2006.01)
H04L 12/56 (2006.01)

Primary Examiner—Joshua Joo
(74) *Attorney, Agent, or Firm*—FERENCE & ASSOCIATES LLC

(57) **ABSTRACT**

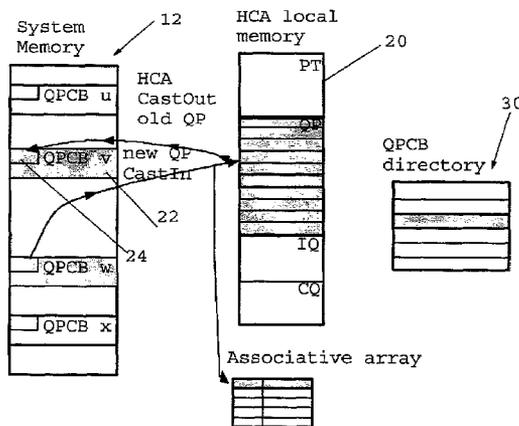
(52) **U.S. Cl.** **709/250**; 709/213; 709/214;
709/215; 709/216; 709/229; 710/15; 710/16;
710/17; 710/18; 710/19; 710/33; 710/52;
710/56; 370/412

A method and apparatus relates to hardware-to-hardware data transmission in computer systems, and in particular, it relates to method and system for operating I/O adapters attaching either one or more computing devices to an I/O periphery, to a network, or to other computing devices. It is proposed to operate a memory local to the network coupling adapter as a cache memory relative to a system memory associated with the one or more computing devices for storing transmission control information.

(58) **Field of Classification Search** 709/218,
709/232, 234, 235, 238, 249, 250, 213–216,
709/229; 710/15–19, 33, 52, 56; 370/412

See application file for complete search history.

12 Claims, 5 Drawing Sheets



US 7,337,240 B2

Page 2

U.S. PATENT DOCUMENTS

6,807,581 B1 * 10/2004 Starr et al. 709/250
6,810,440 B2 * 10/2004 Micalizzi et al. 710/19

FOREIGN PATENT DOCUMENTS

EP 0789302 A 8/1997
EP 1006449 A 6/2000

WO WO 99/03044 1/1999

OTHER PUBLICATIONS

European Search Report dated Jan. 11, 2006, for Application No. EP 01 12 8821.

* cited by examiner

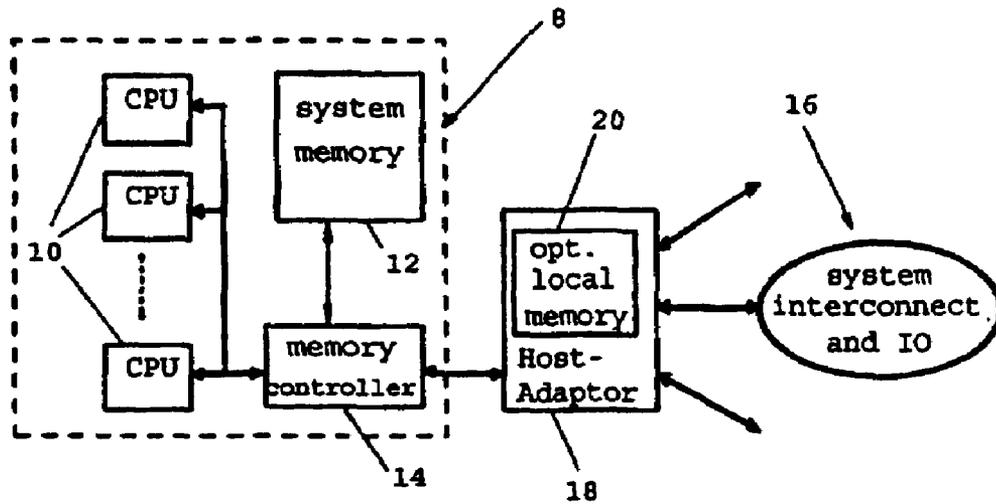


FIG. 1

-- PRIOR ART --

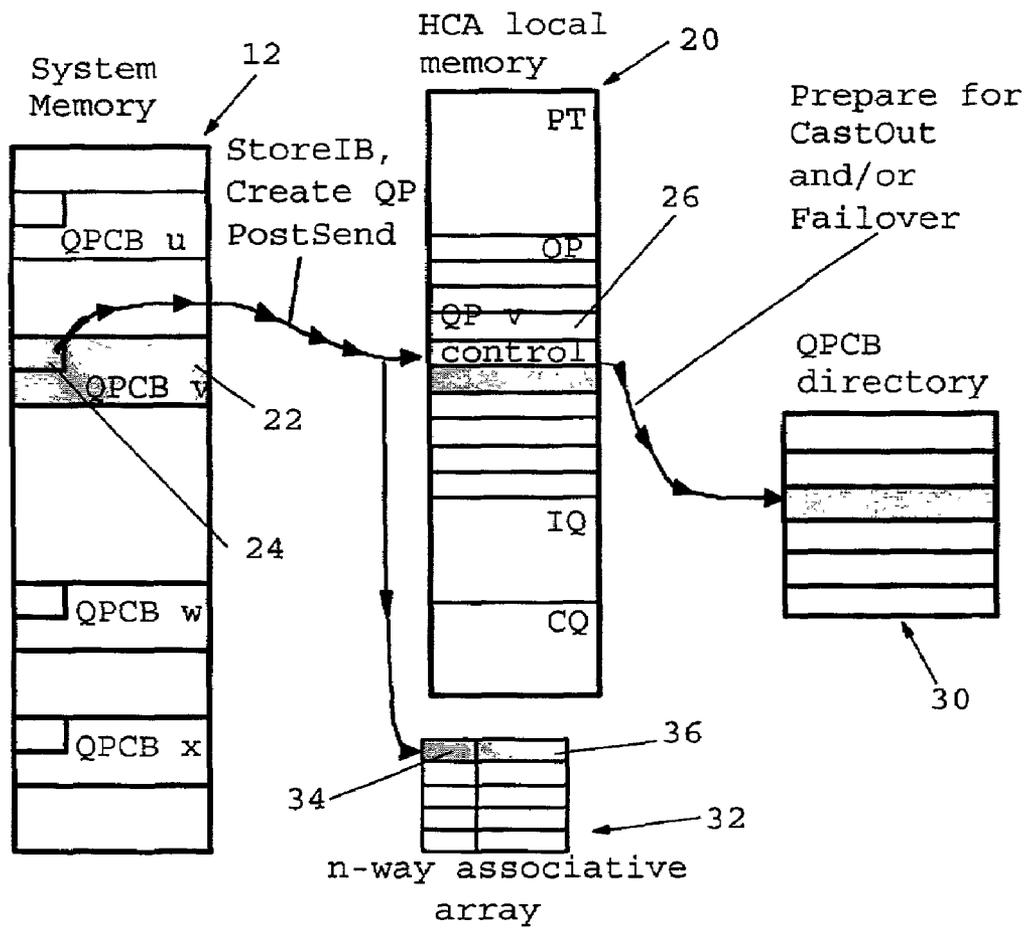


FIG. 2

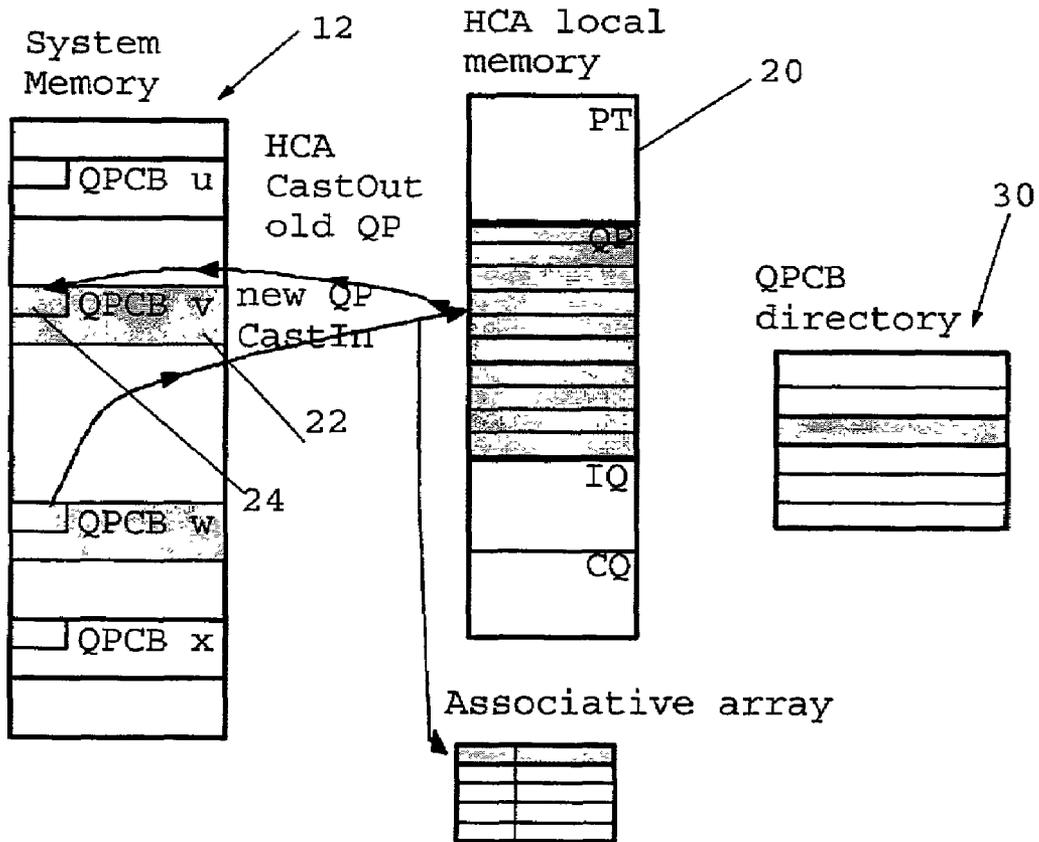


FIG. 3

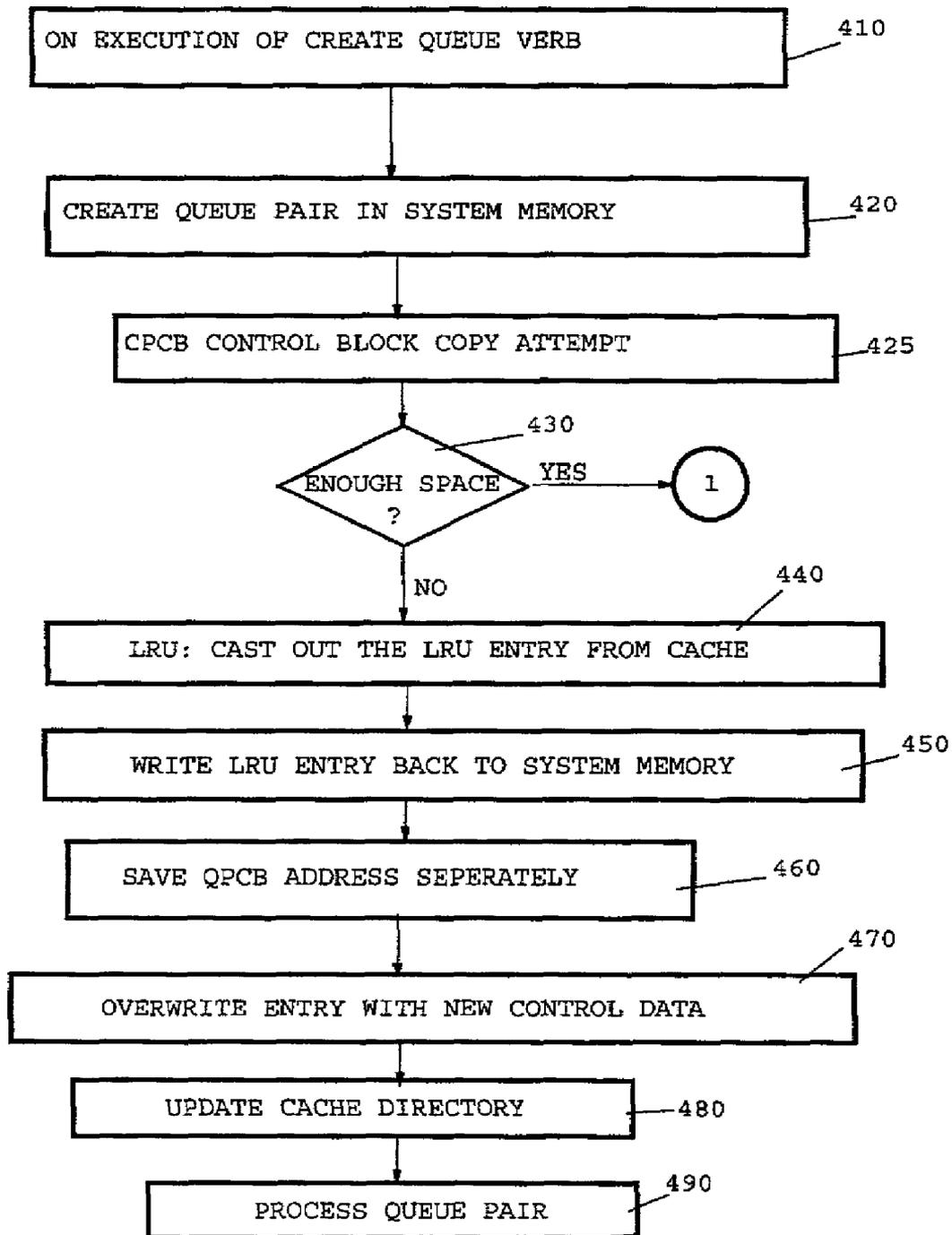


FIG. 4A

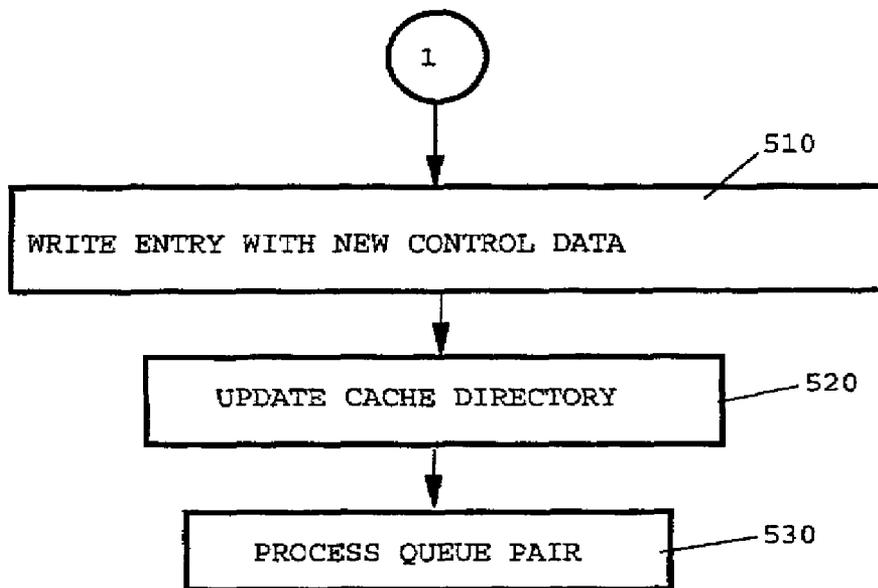


FIG. 4B

VIRTUALIZATION OF I/O ADAPTER RESOURCES

BACKGROUND OF INVENTION

1. Field of the Invention

The subject invention relates to hardware-to-hardware data transmission in computer systems. In particular, it relates to method and system for operating I/O adapters attaching computing devices either to an I/O periphery, a network, or other computing devices.

2. Description and Disadvantages of Prior Art

The area of the invention concerns hardware of computer systems and network components. It deals more particularly with a method to improve the performance of I/O adapters and utilisation of adapter-local resources like memory.

As revealed by a first publication of the InfiniBand Architecture (IBA), the prior art of interconnect technologies have failed to keep pace with the current computer evolution and the increased burden imposed on data servers, application processing and enterprise computing created by the popular success of the Internet.

High end computing concepts such as clustering, fail-safe operations, and 24x7 hour availability demand greater capacity to move data between processing nodes as well as between a processor node and I/O devices. These trends require higher bandwidths and lower latencies, they are pushing more functionality down to the I/O adapters, and they are demanding greater protection, higher isolation, deterministic behavior, and a higher quality of service than it is currently available. InfiniBand helps to achieve the above mentioned aims.

The invention can be advantageously applied with this new InfiniBand technology and thus increases speed of technical evolution.

Although the invention has a quite general scope it will be discussed and set out with reference to a specific prior art hardware-to-hardware data transmission in computer systems. This is a communication between a CPU subsystem **8** and a host adapter **18**, as depicted in FIG. 1 and explained next below.

Today's computer systems (hosts) have a "dense-packed" CPU-memory-subsystem **8** comprising a plurality of CPUs with caches **10**, system memory **12**, memory controller **14**, interconnect logic, etc. Input/output devices, further referred to herein as I/O devices **16** like storage devices, communication networking devices, inter-system connections, etc. are attached via a so-called I/O or host adapter **18**. The host adapter **18** may be connected with some "distance" in terms of access time to the CPU-memory subsystem.

Applications running in the CPUs use specific communication protocols for their connections to said I/O devices **16** and other computer systems accessible via a network.

These protocols, as for example InfiniBand mentioned above, may define that the application can post work requests to the system memory and is enabled to signal the host adapter to process these work requests. This requires, however, that for signaling and control purposes some amount of information has to be transferred from the CPU-memory-subsystem **8** to the host adapter **18**. There are protocols which define very complex tasks for the host adapter to execute in order to perform said processing of the work requests. As it is apparent to a person skilled in the art, a multiple queue processing system is used for processing various incoming requests, in-/outbound data traffic associated with work queues, and system control queues.

In prior art there have been two different types of methods to cope with this problem:

With the first type of methods, the I/O adapter **18** is equipped with local memory **20**, e.g. implemented on-chip or as separate SRAM/DRAM on card or board. The required control information of the posted work requests is stored in this local memory. During processing, the host adapter **18** has fast access to the required information. This approach performs very well, but there are resource restrictions, for example the relatively small maximum number of postable work requests which prevents this prior art approach from scaling up to larger environments. This is primarily due to size limitations of the local memory **20**. A simple up-scaling of the local memory is expensive as it costs too much (e.g. chip area costs or SRAM/DRAM module costs).

With the second type of methods, the I/O adapter is not equipped with local memory. Instead, it contains a small set of registers in logic to hold the required control information of one or more work requests. Processing work requests requires many accesses to system memory. This approach is optimized for cost but would not perform well although it does not imply the resource restrictions of method **1**. This approach would be a significant obstacle for implementing a well performing, fabric-based switching technology such as InfiniBand.

It is thus an objective of the invention to overcome the performance/resource restriction problems as outlined above while concurrently being compatible with the switching technology in general.

SUMMARY OF INVENTION

These objects of the invention are achieved by the features stated in enclosed independent claims to which reference should now be made. Further advantageous arrangements and embodiments of the invention are set forth in the respective subclaims.

According to a primary aspect of the invention a method for improving the performance of a network coupling adapter is disclosed which attaches one or more computing devices via an interconnected memory, to either one of an I/O periphery, a network, or other computing devices. The method is then characterized by the steps of: operating a local memory being associated with the network coupling adapter as a cache memory relative to a system memory, called an interconnected memory, associated with one or more computing for storing transmission control information.

Various other objects, features, and attendant advantages of the present invention will become more fully appreciated as the same becomes better understood when considered in conjunction with the accompanying drawings, in which like reference characters designate the same or similar parts throughout the several views.

BRIEF DESCRIPTION OF DRAWINGS

FIG. 1 is a schematic block diagram showing the structural elements in a prior art computer system being equipped with a host adapter.

FIG. 2 is a schematic block diagram showing the basic structure of an inventional method for caching queue pairs in a first operating state.

FIG. 3 is a schematic block diagram according to FIG. 2 in a second operating state different to that one shown in FIG. 2.

FIGS. 4A and 4B are a schematic diagram showing the basic steps of the associated control flow according to the inventive embodiment shown in FIGS. 2 and 3.

DETAILED DESCRIPTION

The embodiment described next is directed to a design for a transport layer logic of an I/O adapter, i.e., a so-called Host Channel Adapter (HCA) as defined by the InfiniBand Architecture.

The term network and network coupling adapter is to be understood in a very general sense: The network can be for example a Wide Area Network (WAN), a Local Area Network (LAN), or even backplane bus within a PC where the bus participants are interpreted as network attached elements. The term network coupling adapter is thus any hardware device in such hardware structure which interconnects network components.

Said hardware structures include explicitly so-called fabric structures as well as a replacement technology of any kind of conventional bus technology. The expression “fabric” has the general meaning of ‘configuration’. More particularly it is used herein as defined in the Fibre-Channel or the InfiniBand Standards.

Thus, in terms of network topology, it can be considered as an ‘agglomeration, i.e., a ‘cloud’-like structure of point-to-point connections’, in which the bandwidth availability is less restricted compared to conventional bus technology, for example.

As a primary advantage the option is provided to offer a variable number of communication channels without the provision of a large local memory with a fixed size and performance in the network coupling means itself. Thus, the performance of said network coupling means can be easily scaled up according to dynamically changing traffic load without adding large amount of fast and expensive SRAM/DRAM area locally into said device. Thus, the traffic load flexibility is increased significantly.

The above inventive concept can be advantageously used with InfiniBand technology because a modern industry standard is defined therewith which allows to apply said fabric-based concept in the whole range of applications as it was sketched out above.

When the transmission control comprises the processing of address translation, e.g., logical to physical and vice versa—and protection information, e.g., tables then prior art remote processes can be advantageously performed.

When used for connecting a plurality of I/O devices associated with one or more computing devices as described above, the entire I/O periphery can be controlled with less restrictions and better performance according to the invention.

When said transmission control information is bundled per queue or queue pair then the number of cache line transfers to said interconnected memory means for a queue work request is reduced which increases performance and saves bandwidth.

Said cache memory can be configured for special queues not to discard transmission control information after cast out, i.e., after copying said control information back to the main memory. Subsequent repeated cast-in operations can be avoided if the cache line has not been re-used for other control information. A reduced number of cast-in operations and reduced latency improve the processing of queue pairs.

When writing said transmission control information to the local memory only before signaling the completion of a InfiniBand verb, then bandwidth is saved as well.

The inventive method can even be used for providing interprocess communication (IPC) between a plurality of processes associated with one or more computing devices, independent of the underlying hardware structure of the network.

Furthermore it can be advantageously combined with the InfiniBand Architecture specification which was recently published. Amongst the general understanding of a person skilled in the art the following terms are thus used with additional—but not restricted to—particular reference to the InfiniBand Architecture specification, as it was recently published: “Adapters” in the sense of Host Channel Adapters (HCA) or Target Channel Adapters (TCA), “network” including a fabric, or “verbs” which provide an abstract definition of the functionality provided to a host by a Host Channel Interface (HCI).

The basic idea of the invention is, to use system memory as well as host adapter local memory for the transmission control information. The local memory is used like a cache, the system memory hold those work requests, which do not fit into the cache. This allows to provide the performance of implementing local memory only, but overcomes the resource restrictions of the local-memory-only approach.

The InfiniBand Architecture is designed around a point-to-point, switched I/O fabric, whereby end-node devices (which can range from very inexpensive I/O devices like single chip SCSI or Ethernet to very complex host computers) are interconnected by cascaded switch devices.

The invention provides a general means for improving prior art hardware-to-hardware data transmission on a very large range of scales: the invention can thus be advantageously applied to improve data traffic in pure, dedicated network devices like switches and routers, and furthermore, it can be well applied within LAN/WAN-based interprocess communication. The invention’s basic concept is open to integrate any prior art network technology and in particular it can be advantageously applied to techniques such as Ethernet or Fibre Channel.

Thus, according to the invention any hardware-based data transmission like a module-to-module interconnection, as it is typified by computer systems that support I/O module add-in slots or chassis-to-chassis interconnections as they are typified by interconnecting computers, external storage systems or even external LAN/WAN access devices, such as switches, hubs and routers in a data-center environment can be advantageously supported by the invention’s concepts.

In FIG. 2 the system memory 12—depicted left—has a plurality of entries 22 for storing the work request related control information for a particular work queue in a queue pair control block 22, further abbreviated as QPCB, each entry comprising a storage field 24 for storing the control information for it. Other queues are managed in here as well, these are, however not depicted in order to improve clarity.

Among others, the QPCB 22 comprises the following basic data:—queue pair state information—sequence numbers—maximum transfer unit size—destination LID (Local Identifier of connected queue pair)—destination GID (Global Identifier of connected queue pair)—error counters—performance counters. Among others the control field 24 comprises the following basic data:—send and receive queue head and tail pointers—number of associated completion queues—depth of send and receive queues. In the host adapter memory 20 several transmission control blocks, e.g., a protection table PT, the work request queue WQ, with the queue pairs QP, an interrupt queue IQ, and a complete queue CQ, are managed. For each queue a plurality of cache entries 26 is provided for receiving the queue pair ID, i.e., a unique

5

number and the respective control information required for the connecting host adapter to do its job, i.e., route the requested data to the correct network element or I/O device, respectively.

Further, a n-way associative array **32** is provided for storing the queue pair number **34** with the local address **36** in the cache storage **20**, like it corresponds to usual caching techniques. Further, a QPCB directory **30** is provided for storing the queue pair number with the address of the system memory **12**, in order to enable for casting out an entry from the cache memory **20** back into the system memory **12**, when required.

In FIG. **2** enough local memory space is available in the QP area. During operation of the caching mechanism a situation emerges in which there is no free entry in said storage area for the queue pairs. This is depicted in FIG. **3** which has basically the same structure as described above with reference to FIG. **2**.

With general reference to the figures and with special reference now to FIG. **4A** and **B**, the operation of the proposed caching technique will be described in more detail with a sample queue pair as it is defined in the InfiniBand Architecture: a send queue and a receive queue. It should be noted that any other queues required for compatibility with the InfiniBand Architecture, for example, or with other protocols can be managed according to the same principle.

On execution of a CreateQueue verb, e.g., when a queue pair shall be created, step **410**, this is initiated by the CPU-memory-subsystem **8** in FIG. **1**. The respective application which originates the queue pair generation thus triggers that a queue pair control block (QPCB) is built in the system memory **12**, step **420**.

Then the host adapter's cache memory gets a request for storing caching data for the queue pair, i.e., the host adapter **18** gets a door bell signal indicating that the control area **24** of the new control block has to be copied, step **425** to the host adapter. A control logic decides, step **430**, if enough free storage space is available in the cache memory. If not, see the NO-branch **430** of FIG. **4A**, then a classical cast-out/cast-in process takes place: In this situation now the host adapter checks the available storage space and detects that the local cache memory **20** is out of free space. Thus, in a next step **440** one particular queue pair, i.e., only its control information, is cast-out from the local cache memory according to an algorithm, like for example used in conventional caching techniques where for example the least recently used cache entry is overwritten (LRU algorithm). Thus, this entry is written back into system memory, step **450**, and the address of the QPCB **22** is saved in the QPCB directory **30**, step **460**.

Then, in a next step **470** the host adapter **18** writes the new queue pair control block into the respective storage location, for example by simply overwriting the former contents of it.

Finally, the cache directory **32** is updated again, step **480**. Then, the host adapter **18** is enabled to process the new queue pair, step **490**.

When enough space is available in the local cache memory, see the YES-branch of decision **430**, then the sequence of steps for cast-in/cast-out is not required. Instead, see FIG. **4B** now, the respective control information is copied from the system memory **12** to the local memory **20** of the host adapter **18**, step **510**. Thus, only a small fraction of the queue pair data amount, i.e. only the control information is stored in the local cache memory **20**.

Further, said cache directory **32** is updated, as it would be done with usual caching techniques known in prior art

6

within a processor unit, step **520**. Then the request is ready for execution, the queue pairs can be processed, step **530**.

Thus, the invention represents a large step forward to a significantly increased performance in host adapter's work request handling because all transmission control information—which requires only small chip area compared to the total work request data contained in the queue pair—is available immediately where it is required: local to the host adapter. The rest of data which can be sent “through” the host adapter is stored external to the adapter/switching element because it does not carry any routing/switching information. Thus, a person skilled in the art will appreciate that the invention concept can be scaled up and down easily with a small increase or decrease of required chip area needed for the local cache memory **20**—according to the actual requirements present on a given hardware and traffic situation.

In the foregoing specification the invention has been described with reference to a specific exemplary embodiment thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are accordingly to be regarded as illustrative rather than in a restrictive sense.

For example, the way in which the cache memory **20** is operated, can be varied to the different types known in the art, e.g., write back, or write through, etc. .

The invention claimed is:

1. A method for operating a network coupling adapter attaching one or more computing device via an associated interconnected memory to either one of an I/O periphery, a network, or other computing devices, characterized by the steps of:

- operating a local memory on the network coupling adapter for storing transmission control information;
- operating a system memory of the one or more computing device for storing a plurality of entries, an entry of the plurality of entries comprising work related information for a queue or a queue pair and information other than transmission control information, such that the entry stored in the system memory is associated with the transmission control information stored in the local memory, wherein the transmission control information stored in the local memory is received from the entry in the system memory and is used for processing the queue or the queue pair of the entry in the system memory; and

- wherein said step of operating a local memory comprises: determining if there is room in the local memory for storing the transmission control information and a new transmission control block;

- if there is not sufficient room in the local memory for storing the transmission control information and the new transmission control block, moving the transmission control information stored in the local memory to the entry of the system memory and maintaining the association with the entry previously stored in the system memory, and storing the new transmission control block in the local memory; and

- if there is sufficient room in the local memory for storing the transmission control information, storing the transmission control information and the new transmission control block in the local memory.

2. The method according to claim **1** further comprising the steps of using an InfiniBand and Architecture.

7

3. The method according to claim 2 further comprising the steps of using said transmission control for the processing of completion queues.

4. The method according to claim 2 further comprising the steps of using said transmission control for the processing of address translation and protection tables.

5. The method according to claim 2 further comprising the steps of using said local memory for connecting to at least one computer device to a network.

6. The method according to claim 2 further comprising the steps of using said transmission control information for bundled per queue or queue pair.

7. The method according to claim 2 further comprising the steps of configuring said cache memory not to discard transmission control information for particular queues after casting-out.

8. The method according to claim 2 further comprising the step of writing said transmission control information to the local memory only before signaling the completion of a InfiniBland verb.

9. The method according to claim 2 further comprising the steps of using said previous step for connecting a plurality of I/O hardware devices associated with a computing device.

10. The method according to claim 2 further comprising the steps of using said previous step for providing communication channels for interprocess communication between a plurality of processes associated with one or more computing devices.

11. A network coupling element coupling one or more computing devices via an associated interconnected memory to either one of an I/O periphery, a network, or other computing devices characterized by hardware and comprising:

a local memory on the network coupling element being operable as a cache memory, such that transmission control information associated with an entry of a plurality of entries stored in said interconnected memory of the computing device is cached in the local memory and the entry comprising work related information for a queue and information other than transmission control information, wherein the transmission control information stored in the local memory is received from the entry in the interconnected memory and is used for processing the queue of the entry in the interconnected memory;

wherein operating the local memory as a cache comprises: determining if there is room in the local memory for storing the transmission control information and a new transmission control block;

8

if there is no sufficient room in the local memory based on said determining, moving the transmission control information stored in the local memory to the entry of the interconnected memory and maintaining the association with the entry previously stored in the interconnected memory, and storing the new transmission control block in the local memory; and,

if there is sufficient room in the local memory for storing the transmission control information, storing the transmission control information and the new transmission control block in the local memory.

12. A network coupling element for coupling one or more computing devices via an associated interconnected memory to an I/O periphery, and operates by hardware and comprising:

a local memory on the network coupling element being operable as a cache memory, such that transmission control information associated with an entry of a plurality of entries stored in said interconnected memory of the computing device is cached in the local memory and the entry comprising work related information for a queue and information other than transmission control information, wherein the transmission control information stored in the local memory is received from the entry in the interconnected memory and is used for processing the queue of the entry in the interconnected memory;

wherein operating the local memory as a cache comprises: determining if there is room in the local memory for storing the transmission control information and a new transmission control block;

if there is no sufficient room in the local memory based on said determining, moving the transmission control information stored in the local memory to the entry of the interconnected memory and maintaining the association with the entry previously stored in the interconnected memory, and storing the new transmission control block in the local memory; and,

if there is sufficient room in the local memory for storing the transmission control information, storing the transmission control information and the new transmission control block in the local memory,

wherein the network coupling element operates either as a Host Channel Adapter or a Target Channel Adapter being operable according to InfiniBand Architecture.

* * * * *