10338347 .110402

Attorney Docket No. NWISP024

PATENT APPLICATION

METHODS AND APPARATUS FOR MANAGING PROBE REQUESTS

Inventor(s): David B. Glasco
10337 Ember Glen Drive
Austin, TX 78726
Citizen of the U.S.

Assignee: Newisys, Inc.
A Delaware corporation

BEYER WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, California 94704-0778
(510) 843-6200

METHODS AND APPARATUS FOR MANAGING PROBE REQUESTS

CROSS-REFERENCE TO RELATED APPLICATIONS

The present application is related to filed U.S. Application No. 10/106,426 titled Methods And Apparatus For Speculative Probing At A Request Cluster, U.S. Application No. 10/106,430 titled Methods And Apparatus For Speculative Probing With Early Completion And Delayed Request, and U.S. Application No. 10/106,299 titled Methods And Apparatus For Speculative Probing With Early Completion And Early Request, the entireties of which are incorporated by reference herein for all purposes. The present application is also related to filed U.S. Application Nos. 10/157,340, 10/145,439, 10/145,438, and 10/157,388 titled Methods And Apparatus For Responding To A Request Cluster by David B. Glasco, the entireties of which are incorporated by reference for all purposes. The present application is also related to concurrently filed U.S. Application No. __/_______ (Attorney Docket No. NWISP025) with the same title and inventor, the entirety of which is incorporated by reference herein for all purposes.

BACKGROUND OF THE INVENTION

1. Field of the Invention.

The present invention generally relates to accessing data in a multiple processor system. More specifically, the present invention provides techniques for improving data access efficiency while maintaining cache coherency in a multiple processor system having a multiple cluster architecture.

2. Description of Related Art

Data access in multiple processor systems can raise issues relating to cache coherency. Conventional multiple processor computer systems have processors coupled to a system memory through a shared bus. In order to optimize access to data in the system memory, individual processors are typically designed to work with cache memory. In one example, each processor has a cache that is loaded with data that the processor frequently accesses. The cache is read or written by a processor. However, cache coherency problems arise because multiple copies of the same data can co-exist in systems having multiple processors and multiple cache memories. For example, a frequently accessed data block corresponding to a memory line may be loaded into the cache of two different processors. In one example, if both processors attempt to write new values into the data block at the same time, different data values may result. One value may be written into the first cache while a different value is written into the second cache. A system might then be unable to determine what value to write through to system memory.

A variety of cache coherency mechanisms have been developed to address such problems in multiprocessor systems. One solution is to simply force all processor writes to go through to memory immediately and bypass the associated cache. The write requests can then be serialized before overwriting a system memory line. However, bypassing the cache significantly decreases efficiency gained by using a cache. Other cache coherency mechanisms have been developed for specific architectures. In a shared bus architecture, each processor checks or snoops on the bus to determine whether it can read or write a shared cache block. In one example, a processor only writes an object when it owns or has exclusive access to the object. Each corresponding cache object is then updated to allow processors access to the most recent version of the object.

Bus arbitration is used when both processors attempt to write the same shared data block in the same clock cycle. Bus arbitration logic decides which processor gets the bus first. Although, cache coherency mechanisms such as bus arbitration are effective, using a shared bus limits the number of processors that can be implemented in a single system with a single memory space.

Other multiprocessor schemes involve individual processor, cache, and memory systems connected to other processors, cache, and memory systems using a network backbone such as Ethernet or Token Ring. Multiprocessor schemes involving separate computer systems each with its own address space can avoid many cache coherency problems because each processor has its own associated memory and cache. When one processor wishes to access data on a remote computing system, communication is explicit. Messages are sent to move data to another processor and messages are received to accept data from another processor using standard network protocols such as TCP/IP. Multiprocessor systems using explicit communication including transactions such as sends and receives are referred to as systems using multiple private memories. By contrast, multiprocessor system using implicit communication including transactions such as loads and stores are referred to herein as using a single address space.

Multiprocessor schemes using separate computer systems allow more processors to be interconnected while minimizing cache coherency problems. However, it would take substantially more time to access data held by a remote processor using a network infrastructure than it would take to access data held by a processor coupled to a system bus. Furthermore, valuable network bandwidth would be consumed moving data to the proper processors. This can negatively impact both processor and network performance.

Performance limitations have led to the development of a point-to-point architecture for connecting processors in a system with a single memory space. In one example, individual processors can be directly connected to each other through a plurality of point-to-point links to form a cluster of processors. Separate clusters of processors can also be connected. The point-to-point links significantly increase the bandwidth for coprocessing and multiprocessing functions. However, using a point-to-point architecture to connect multiple processors in a multiple cluster system sharing a single memory space presents its own problems.

Consequently, it is desirable to provide techniques for improving data access and cache coherency in systems having multiple clusters of multiple processors connected using point-to-point links.

SUMMARY OF THE INVENTION

According to the present invention, methods and apparatus are provided for increasing the efficiency of data access in a multiple processor, multiple cluster system. Mechanisms for reducing the number of transactions in a multiple cluster system are provided. In one example, probe filter information is used to limit the number of probe requests transmitted to request and remote clusters.

In one embodiment, a computer system is provided. The computer system includes a home cluster having a first plurality of processors and a home cache coherence controller. The first plurality of processors and the home cache coherence controller are interconnected in a point-to-point architecture. The home cache coherence controller is configured to receive a probe request and probe one or more selected clusters. The one or more clusters are selected based on the characteristics associated with the probe request.

In another embodiment, a method for managing probes is provided. A probe request is received at a home cache coherence controller in a home cluster. The home cluster includes a first plurality of processors and the home cache coherence controller. The first plurality of processors and the home cache coherence controller are interconnected in a point-to-point architecture. One or more clusters are selected for probing based on the characteristics associated with the probe request. The one or more clusters are probed.

A further understanding of the nature and advantages of the present invention may be realized by reference to the remaining portions of the specification and the drawings.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.