

The Load Balancing System Design of Service Based on IXP2400

Yi Shijun^{1, a}, Jing Xiaoping^{1, b}

¹ Chengdu Electromechanical College, China

^aysj1967@yahoo.com.cn, ^bvoicent@163.com

Keywords: Load balance, Service performance, IXA, IXP2400, Microengine

Abstract. This paper proposes a new method of designing load balancing system. The design of the system is based on the performance of hosting service and the switching technology and makes use of the highly flexible programmability and the powerful function of processing network packet of IXP2400 network processor. This method greatly accelerates the processing speed of the system and owing to the advanced technology of network processor it makes the load-sharing system excellent in usability and practicality.

Introduction

A. Questions

The rapid increase in network access services makes a high demand on the server's concurrent access ability. This high intensity of data flow and calculations makes single device simply can not undertake [1]. On the other hand, it's also an urgent task to accomplish a reasonable allocation of the volume of business between multiple network service providers which have the same function and data-switching equipment and avoid getting a piece of equipment too busy, while other equipments can not exert fully the processing capacities.

How to achieve a reasonable distribution of the volume of business among multiple network loads that perform same function has become a problem, and load balancing mechanism therefore came into being. To solve this problem, this paper designs a load balancing system using IXP2400 [2] on the basis of combining the host service performance and switching technology.

B. The load balance technology combined with hosting performance

The most important traditional load balancing techniques include: Round Robin Algorithm (Round Robin), the ratio method (Ratio), response speed algorithm (Response Time), at least join algorithm (Least Connection) and so on [3].

These traditional load balancing approaches have some defects [4], such as they can not guarantee the reliability of the system, and load sharing are not necessarily rational. The traditional load balancing algorithm can not judge the performance of specific services. For example, if the services of a server crash, with the above mechanism the server ping can not detect whether the services of this server can work or not, it will still sent a steady stream of requests to the server, which will lead to service failure.

In addition, there are load balancing cluster technology and a dedicated server and other technical programs, but they still have the defects of high cost, limited function and other shortcomings [5].

The proposed load balancing system based on service performance and the switching technology is a mechanism specially designed for server load sharing to solve the problems caused by traditional methods, improve service reliability, and distribute load reasonably among application servers.

Its working principle is: first each server has a mechanism to determine the local service level performance and availability, and the server sends performance parameters at regular time to the load balancing system (hereinafter referred to as system). The server performance data is updated regularly by the balancing system according to the parameters. The system only has one external IP address, all the servers sharing same service assignments are in a same group and they have internal IP. When the new service request arrives, the system will locate and connect a server which is undertaking the lightest load according to each server's performance, while preserving the source IP and target IP. All the follow-up packages of a same session will be transmitted by the second floor exchange module without load-sharing process.

C. IXP2400 network processor

IXP2400 is one member of Intel Internet Exchange Architecture (IXA) [6] second-generation

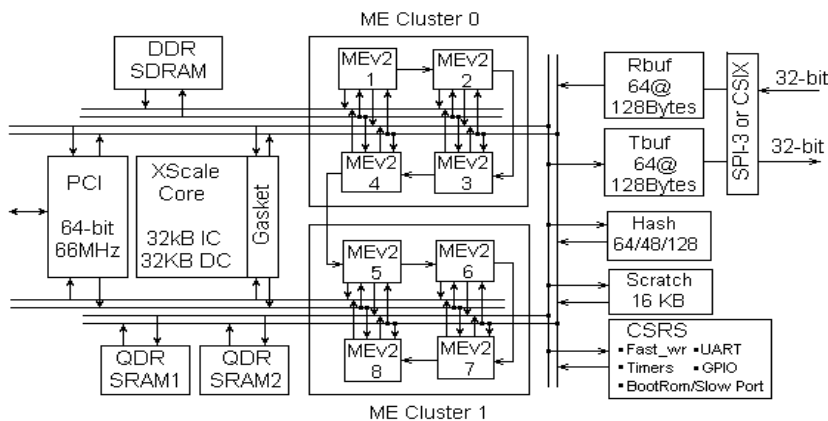


Fig 1 IXP2400 Architecture

network processor family that is designed to meet the respective requirements of CPE, access, edge, and core market segments. It integrates an Intel XScale core with eight multi-threaded, independent 32-bit microengines that cumulatively enable a total of 4.8 gigabit-operations per second.

Intel IXA is a network processing architecture that has two defining elements [7]:

- Microengine technology: a subsystem of programmable, multi-threaded RISC microengines that enable high-performance packet processing in the data plane through Intel Hyper Task Chaining.
- Intel XScale technology: providing the highest performance-to-power ratio in the industry, with performance up to 1,000 MIPS and power consumption as low as 10 MW, for low-power, high-density processing of control-plane applications.

The Load Balancing System Design Based on IXP2400

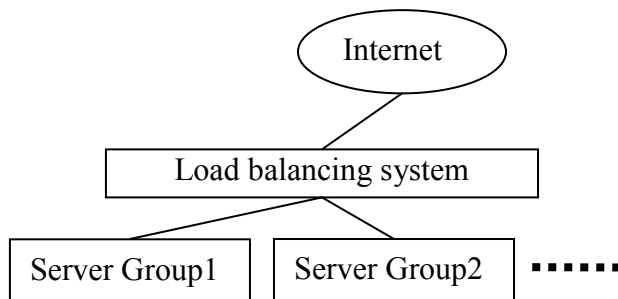


Fig 2 Load balancing system deployment

The load balancing system is divided into two parts [8]: hardware and software. The system aims at large-scale application load sharing, and its deployment is as shown in Figure 2.

A. The Design of hardware

IXP2400 is a second generation network processor under Intel IXA framework, the design based on IXP2400 is relatively simple. The hardware design

of IXP2400-based load balancing system is as shown in Figure 3.

This is a multi-layer switch hardware configuration structure based on dual IXP2400 (respectively, Ingress and Egress). The hardware framework transmit data packet with a store-transmit method, achieving load sharing with queuing techniques. The data packet is sent to the corresponding physical port after processed by the Ingress and Egress network processor.

In figure 3, each of the two IXP2400 network processor has eight micro-engines, each micro-engine supports eight hardware threads (Context), and for each hardware thread we can assign different tasks (i.e., allocation of resources). From Figure 3 it can be seen that based on this design the processor Ingress is used to handle the Ethernet IP packets entering into the system, and Egress is used to send data packets to the physical port.

Switch Fabric is a high-speed data exchange platform, which can connect many IXP network processing devices and it directly transmits data from one port to the designated port through a high-speed switching board.

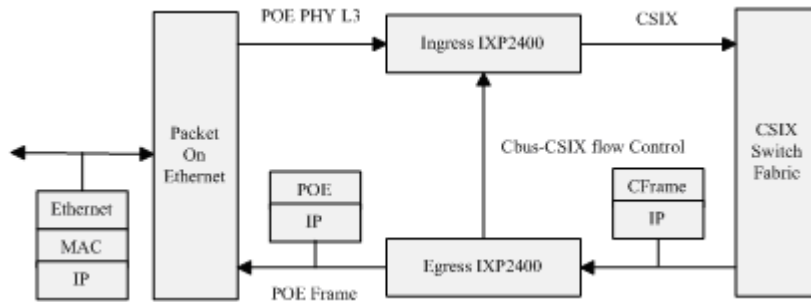


Fig 3 The Hardware Architecture of NIDS

The System has four Ethernet cards supporting 1Gbps speed, which provides input and output of Ethernet network data.

Flow Control is a control plane processor. In this design, on the whole, it is divided into control plane and data plane. Control plane handles only abnormal data and other non-forward data; the data plane provides general data processing.

B. The Design of Software Architecture

1) The key data structure design

The key data structure for load balancing judgment of this system is determined by the system taking into consideration of the performance of each group of hosts. Its structure design is as shown in Table 1.

Table 1 The key data structure

Field	Type	Size	Help
Number.	Byte	8bit	Number of server
Server IP	Byte	4byte	The internal IP address of internal server
Performance index 1-10	Short byte	8bit	Default 0
Performance index 1-10 weight	Short byte	8bit	Default 0
Comprehensive performance index	Short byte	8bit	
Whether the server is working	Bool	1bit	Working 1, not working 0
Performance update overtime	Short byte	8bit	If there is more than 10 times of update overtime for the performance index, then the server is in a non-normal operation, until the index gets updated.

Server performance index is indicated by comprehensive performance indexes ranging from 1 to 10, up to 10 individual performance indexes. Each index has a certain weight, 10 indexes have a weight of 100% (see Equation 1), Set the performance index as X, and the weight as λ , then the comprehensive performance index is as in Equation 2.

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_9 + \lambda_{10} = 100 \quad \text{---Eq.1}$$

$$\lambda_1 * X_1 + \lambda_2 * X_2 + \lambda_3 * X_3 + \lambda_4 * X_4 + \dots + \lambda_{10} * X_{10} = \lambda \quad \text{---Eq.2}$$

Weight λ is collected dynamically by each host's sending IP packets to the load balancing system according to their own service performance. The value of performance index, up to 10, is determined according to the host performance. Balanced system chooses the most suitable server according to the weight λ . The assumed choosing method here is: the better the performance of the server, the greater

the index value. Under the premise that all servers are working normally together, a server that has the largest performance value can be spotted in a rotative searching process. And connection with it can be established.

2) Process Design

The load balancing system work flow shown in Figure 4.

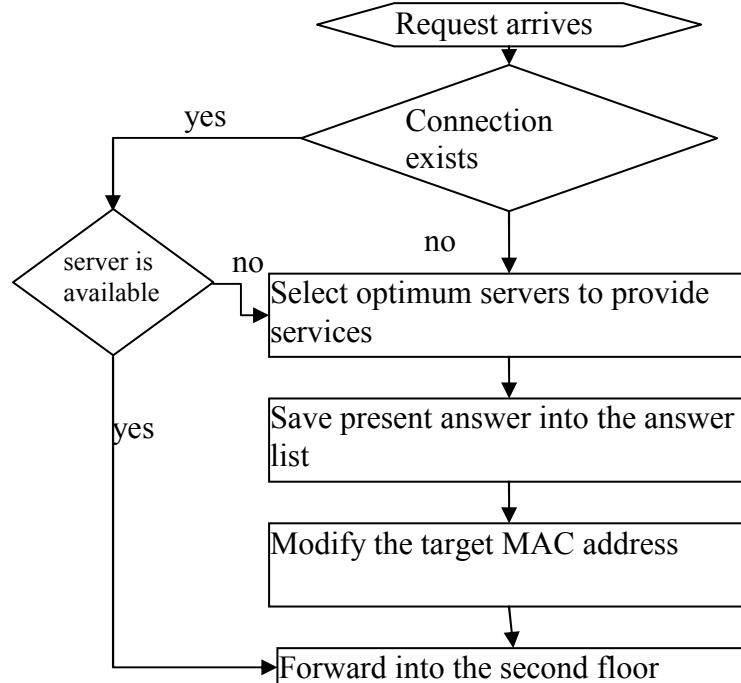


Fig 4 Load balancing system flowchart

Figure 4 shows the processing algorithm of the load-sharing system in the process of sharing data stream. Any time a service request comes, the system first determines whether the connection from the IP exists, and if not, it goes to the server performance list to find an optimum server, saves the answer, changes MAC address, and forwards to enter the second stage transmit; if the connection exists, the system determine whether the current server is available, if so it can forward directly into the second stage transmit; if it is not available, forward to new choice.

3) The implementation based on IXP2400 network processor

In concordance to IXA system structure, the characteristics of IXP2400 network processor and the demand of software design, the system design is divided into three layers: the control plane, data plane and management plane.

1).The control plane operates on the IXP2400 Xscale core processor. It is responsible for maintaining various tables of information and the protocol stack. It deals with "special" IP data packets (such as packet regularly asking about "health" status of application servers and its response packet and other packets).

2).Data plane operates on micro-engines. It completes the fast processing and transmitting of normal data packet.

3).Management plane is either operated by the Xscale processor, or is connected to external general processor. It completes the user login and management, load-sharing strategic management, workflow management, queue scheduling strategic management and log management.

Since the data plane completes the main function of managing data packets and load balancing, the following part focus on the analysis of its design and implementation process. Figure 5 shows the software modules flow diagram running on the micro- engine.

1).POE_RX: Data packet from Ethernet Microblock. When a data packet arrives from Ethernet port, it assembles in IXP2400 memory.

2). IP packet classification --- IP packets received needs to be classified and processed

3).Load balancing system--This module implements load balancing algorithm. It deals with data packet in a process shown in Part2.2.2. At the same time, the module is also responsible for maintaining the availability status of all application servers, including the number of physical links, the health status of the application server hosting and application services.

4). Queue management and queue scheduling --- these two modules are in charge of into- the team and out-of the team operation, and the prescriptive queue scheduling algorithm.

5). CSIX Tx ---It adds data packet from the queue management module with the relevant information from the precedent procedure(including classified information, traffic information, load sharing information) and sent them to the Switch Fabric to enter the next IXP2400 network processor.

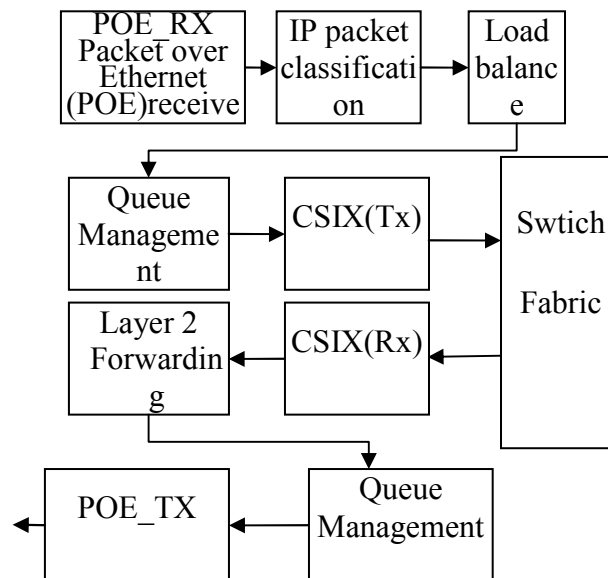


Fig 5 Micro-engine flow diagram of software modules

6). CSIX Rx ---make preparation for sending the data. It's a reverse process of CSIX Rx.

7). Second-stage transmit --- the processed data is sent to second stage for transmitting.

Summary

Faced with ever expanding information access to network, some hot issues demand prompt solutions in the information society: reasonable load for network site servers, the scalability and high availability of servers and bettering the service for users. The paper proposes a solution of using load balancing system based on the performance of hosting service and the IXP2400 network processor. This solution takes full advantage of processor's fast and efficient packet, making the server group serve users in a better way. Meanwhile, network processor's highly flexible programmability and scalability shorten the time of developing the system and enhance the utility and marketability of the system.

References

- [1] <http://baike.baidu.com/view/51184.htm>NAS.
- [2] Intel® IXP2400/IXP2800 Network Processor Programmer's Reference Manual -----Intel Corporation, 2003
- [3] http://www.yesky.com/20010626/187006_3.shtml
- [4] <http://www.51kaifa.com/html/js wz/200511/read-3191.htm>
- [5] WU Yu, Design of Load Balancing System Based on Fourth Level Interchange Using IXP2400 Network Processor[J].Application Research Of Computers,2005,9:253-255.
- [6] IXA Portability Framework Developer's Manual-Intel Corporation, 2003
- [7] Microengine C Compiler Language Support Reference Manual-Intel Corporation, 2003
- [8] ZHONG Ting,Study on fast packet filter under network processor IXP2400[J],Computer Applications,Dec./2005,11:2569.