Review on a Hash-list Based Server Architecture for Speed Optimization on FPGA

Ms. Shifali R. Pande
M. Tech student
S. D. College of Engineering,
Selukate, Wardha.
(shifali137@gmail.com)

Prof. R. H. Deshmukh,
Assistant Professor
S. D. College of Engineering,
Selukate, Wardha.
(rajal.deshmukh@gmail.com)

Abstract:- In Cloud Computing systems, web serversare the major channel of cloud computing. It is necessary to obtain high throughput and reduce power consumption for improving the performance of webservers. In this paper, we propose a novel architecture of web server based on FPGA which places the HTTP data sending part of the web service protocols processing into hardware for acceleration and processing on 8051 for low cost. Results show that our proposed architecture brings higher throughput which achieves our requirement, but it costs fewer resources than existing hybrid-based web servers. Cloud Computing distributes computation task on the resource pool which consists of massive computers, accordingly, the application systems can gain the computation strength, the storage space and software service according to its demand. As Cloud Computing provides the services through networks, design and research of web server are important. However, the massive TCP/IP processing of CPUs restricts the performance improvement. In recent years, many effective web servers have been proposed and implemented based on the original models. There are three implementations of web server: software-based, hardware-based and hybrid. The software-based solution is generally realized by a network processor or microprocessor running a real-time operating system. This method is flexible and available, but lack of scalability to Ethernet. The hardware-based designs usually offer better performance; however, the hardware system is difficult to expand. We provide a hardware-based approach in order to get a small and low-power web server. Soft-core processors are embedded. But they consume too many resources and they are difficult to control for realizing their own protocol. Despite the hybrid-based web servers are the trends to reach the requirements of cloud computing.

Index Terms: Field Programmable Gate Arrays (FPGA); Hyper Text TransferProtocol (HTTP); Application-specific integrated circuit (ASIC);Key-value stores (KVS)

*****

1. INTRODUCTION

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing hence called as "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to an application-specific integrated circuit (ASIC). FPGAs contain an array of programmable logic blocks, and a hierarchy of reconfigurable interconnects that allow the blocks to be wired together, like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. As FPGA designs employ very fast I/O and bidirectional data buses, it becomes a challenge to verify correct timing of valid data within setup time and hold time. The FPGA grows from programmable read-only memory (PROM) and programmable logic devices (PLDs). PROMs and PLDs both had the option of being programmed in batches in a factory or in the field (field-programmable). However, programmable logic was hard-wired between logic gates. A recent trend has been to take the coarse-grained architectural approach a step further by combining the logic blocks and interconnects of traditional FPGAs with embedded microprocessors and related peripherals to form a complete "system on a programmable chip".. The extensible processing platform enables system architects and embedded software developers to apply a combination of serial and parallel processing to their embedded system designs, for which the general trend has been to progressively increasing complexity.

Now a web server, it is a computer system that processes requests via HTTP, the basic network protocol used to distribute information on the World Wide Web. This refers to the entire system, or specifically to the software that accepts and supervises the HTTP requests. The primary function of a web server is to store, process and deliver web pages to clients. The communication between client and server takes place using the Hypertext Transfer Protocol (HTTP). Pages delivered are most frequently HTML documents, which may include images, style sheets and scripts in addition to text content. A user commonly a web browser or web crawler, initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource and with an error message if unable to do so. Web servers are not only used for

serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may be used as a part of a system for monitoring and/or administering the device. This usually means that no additional software has to be installed on the client computer, since only a web browser is required as it is not installed in any system previously.

A server is a computer program or a device that provides functionality for other programs or devices, called "clients". The architecture is called the client–server model, and a single overall computation is distributed across multiple processes or devices. A single server can serve multiple clients, and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device. The typical servers are database servers, file servers, mail servers, print servers, web servers, game servers, and application servers. Client–server systems are today most frequently implemented by and often identified with the request–response model. A client sends a request to the server, which performs some action and sends a response back to the client, typically with a result or acknowledgement. Designating a computer as "server-class hardware" implies that it is specialized for running servers on it. The server is part of the client–server model, where a server serves data for clients. The nature of communication between a client and server is request and response.

With the increasing volume of communication and computation on network, Cloud Computing as a kind of novel business computation model is proposed. As Cloud Computing provides the services through networks, design and research of web server are important. So to work on the cloud computing in recent years, many effective web servers have been proposed and implemented based on the original models. But the three implementations of web server are as software-based web server, hardware-based web server and hybrid web server. Software-based solution is generally realized by a network processor running a real-time operating system. The possible optimization is to improve the efficiency to access the website. Hardware-based designs usually offers better performance, but the hardware system is difficult to expand. Hybrid implementations, which are usually embedded systems realized on FPGA, provide scalability and flexibility while maintaining performance. But they consume too many resources and they are difficult to realize.

2. LITERATURE REVIEW

[1]In this paper, the author aim at architecting high performance and efficient KVS platforms, and start with a rigorous architectural characterization across system stacks over a collection of representative KVS implementations. Distributed in-memory key-value stores (KVS), such as memcached, have a data serving layer in modern Internet-oriented datacenter infrastructure. This performance and efficiency directly affect the Quality of System of web services and the efficiency of datacenters.Hardware-centric research has explore specialized platforms including FPGAs for KVSs and the results demonstrated as an order of magnitude increase in throughput and energy efficiency over stock memcached. detailed full-system characterization not only identifies the critical hardware/software ingredients for high-performance KVS systems, but also leads to guided optimizations on the top a recent design to achieve a record setting throughput of million requests per second (MRPS) on a single commodity server. The author also show that an important building block for large-scale Internet services, key-value stores affect both the service quality and energy efficiency of datacenter-based services. This paper evaluates and improves the scaling and efficiency of both legacy and cutting-edge key-value implementations on commodity servers.Beyond optimizing to achieve the record-setting MRPS performance and KRPS/watt energy efficiency on commodity dual-socket KVS system, this paper sets a principle for future throughput-intensive architectural support for high performance KVS platforms.

[2]In this paper, they propose a novel architecture of web server based on FPGA (Field Programmable Gate Arrays) which places the HTTP data sending part of the web service protocols processing into hardware for acceleration and other processing on 8051 soft-core for low cost. As in Cloud computing systems, web servers the major channel of cloud computing are required with high performance. It is necessary to obtain high throughput and reduce power consumption for improving the performance of webservers. These results show that this proposed architecture brings higher throughput which achieves up to Mbps over Gbps Ethernet, but it costs fewer resources than existing hybrid-based web servers. Besides this they confirm that hybrid implementation is more suitable for cloud computing. They also show that this achieves better performance than other software-based web servers as the transfer size is larger. However, for reducing the storage pressure operations on the data it has to be optimized.

[3]This Paper focuses on a portable low power automated web server in a FPGA platform by accomplish the Network on chip concept. This web server has the capability to adapt multipurpose operation, a client can share the web content not only the HTML files but also can control a device operated in a remote place through this web server. The demand for high speed portable networks forced compulsory optimization in various aspects. A common web server is to act as a central control point which

responds to the user request, and employing a high sophisticated computer for this purpose is not obligatory. The HTML files with JPEG, bmp images can be served through triple speed reconfigurable Ethernet port between FPGA server and client.The hardware design board of FPGA is configured using high level VHDL Language with low power elliptic curve cryptography. User authentication logic is developed in C++ language to encrypt and decrypt the HTML contents. The web server logic for HTML file sharing and device control module is merged into a single entity, from which a client user can select any mode as per requirement.

[4]In this paper, the methodology to port open source Embedded Linux on Virtex II Pro development environment using PowerPC is explored and presented.Another aspect that is explored in this work is to configure the target board as Web Server. View-Control-Model architecture is used to implement the Web Server.The IP assigned to the target board was generated by the network server; hence the target board can be accessed remotely from the system which lies in the same local network. However, if public IP assigned to the target board then it is possible to access the board from any corner of the world through internet.Linux is used because of its strong level of security concurrency in operation similar to RTOS and most important is that it is open source. The use of Linux is royalty free. At the same time availability of the code, gives developers a 360° freedom in development.

[5] In the paper, they present a novel highly-scalable server architecture that seamlessly integrates variable combinations of general purpose CPUs, embedded CPUs, FPGAs, and GPUs. Embedded CPUs based on the latest ARM Cortex-A15 devices with integrated embedded GPUs are combined with FPGA-based reconfigurable SoCs, which can be used for application-specific hardware acceleration. The main challenge on the way is to make system capable of achieving billion and billions calculations per second computing in a significant improvement in power efficiency.In this paper, the architecture of the RECS®|Box system has been presented. The RECS®|Box system is a scalable cluster server system designed to support high density installations. Apart from the high density, the system is designed in a modular fashion, as it allows the seamless integration of heterogeneous compute boards.

3. MOTIVATION

As the performance of key-value stores affects the Quality of System of web services whose efficiency will affects the datacenter. So as to get the desired requirement at less latency time we are using the new architecture.

- A need of improving the performance and efficiency of KVSs.
- The delay of the KVP based system can be optimized and motivates us to develop a new architecture which will improve the speed of the entire server system.
- The KVS use a large latency time and its efficiency is less.

4. PROBLEM DEFINITION

The existing approach is limited by the architecture of sending requests to the server, the more the number of requests, higher the delay. Thus, our problem definition is to

- Reduce the number of requests given to the server with the help of a modern day data structure.
- Select a custom developed Hash List structure, which will store all the variables in one list and send this list as a single request.
- Result in reduced delay and it will improve the speed of the entire server architecture.

5. CONCLUSION

Proposed work will result in increase in server request per second and increase efficiency of the server. This architecture can improve the performance as the existing one as it requires the system to send and receive requests continuously. In this proposed architecture, all requests data collect once in the structure and passes this to the server. Thereby, the numbers of requests are reduced and the speed of the system also improved.

6. REFERENCE

[1] *Sheng* Li, Hyeontaek Lim, Victor W. Lee, Jung Ho Ahn, AnujKalia, Michael Kaminsky, David G. Andersen, Seongil O, Sukhan Lee, PradeepDubey Intel Labs, Carnegie Mellon University, Seoul National University "Architecting to Achieve a Billion Requests per Second Throughput on a Single Key-Value Store Server Platform" IEEE-2015

[2] Junjie Zhang and JinjinTian "Design and Implementation of an Efficient Web Server Based on FPGA" IEEE-2012

[3] Dr.K.V.Karthikeyan, K.Gomathy "FPGA Implementation of Modified Secured Web Server" IJET-2013

[4] Rajendra Patel, ArvindRajawat and R.N.Yadav "Remote Access of Peripherals using Web Server on FPGA Platform"IEEE-2010

[5] René Griessl,MeysamPeykanu, Jens Hagemeyer, Mario Porrmann, Stefan Krupop, Michavordem Berge, Thomas Kiesel and Wolfgang Christmann "A Scalable Server Architecture for Next-Generation Heterogeneous Compute Clusters" IEEE-2014

[6] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, "Workload analysis of a large-scale key-value store" in SIGMETRICS, 2012.

[7] A. Belay, G. Prekas, A. Klimovic, S. Grossman, C. Kozyrakis, and E. Bugnion, "IX: A protected dataplane operating system for high throughput and low latency," in OSDI, 2014.

[8] M. Blott, K. Karras, L. Liu, K. Vissers, J. Bär, and Z. István, "Achieving 10Gbps line-rate key-value stores with FPGAs," in HotCloud, 2013.

[9] A. Gutierrez, M. Cieslak, B. Giridhar, R. G. Dreslinski, L. Ceze, and T. Mudge, "Integrated 3D-stacked server designs for increasing physical density of key-value stores," in ASPLOS, 2014.

[10] R. Kapoor, G. Porter, M. Tewari, G. M. Voelker, and A. Vahdat, "Chronos: Predictable low latency for data center applications," in SOCC, 2012.

[11] Y. Mao, E. Kohler, and R. T. Morris, "Cache craftiness for fast multicore key-value storage," in EuroSys, 2012.

[12] C. Mitchell, Y. Geng, and J. Li, "Using one-sided RDMA reads to build a fast, CPU-efficient key-value store," in USENIX ATC, 2013.

[13] Yamaguchi, Y., Maruyama, T., and Konagaya, A. 'High Speed Homology Search with FPGAs', Proceedings of the Pacific Symposium on Biocomputing, pp.271-282, 2002.

[14] Benkrid, K., Liu, Y., and Benkrid A., 'Design and Implementation of a Highly Parameterised FPGA-Based Skeleton for Pairwise Biological Sequence Alignment'. FCCM'07, pp. 275-278, 2007.