ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

Design of an 8-Bit Array for Client Server Communication for Hash List Architecture

Shifali R. Pande, Prof. R. H. Deshmukh

M. Tech student, Dept. of ECE, SDCE, Wardha, Maharashtra, India

Assistant Professor, Dept. of ECE, SDCE, Wardha, Maharashtra, India

ABSTRACT: - In Cloud computing systems, web servers are the major channel of cloud computing. It is necessary to obtain high throughput and reduce power consumption for improving the performance of web servers. Results show that our proposed architecture brings higher throughput which achieves our requirement, but it costs fewer resources than hybrid web servers. 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 on an operating system. This method is flexible and available. 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. Despite the hybrid-based web servers are the trends to reach the requirements of cloud computing.

KEYWORDS: Field Programmable Gate Arrays (FPGA); Hyper Text Transfer Protocol (HTTP); Application-specific integrated circuit (ASIC); Key-value stores (KVS)

I. 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). 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).

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. 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 pro-vides 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

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. 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 on an 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.

II. RELATED WORK

[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 web servers. 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.

[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 billions 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

III. 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 KVS.
• 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.

IV. 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.

V. OBJECTIVES

The basic objective of the work is

- To develop a hash list based server architecture.
- To improve the speed of client server communication using Hash List Architectures.
- To reduce the number of requests to the server.
- To increase the efficiency.
- To make efficient and low cost web server architecture for implementation on FPGA.

VI. PROPOSED WORK

In this paper, we are taking two steps. Firstly we generate an array for storing the keys and values in the server machine. And the second step is to design a model showing the communication between the client and server machines. The step wise flowchart showing the design steps is shown in below figure 1.

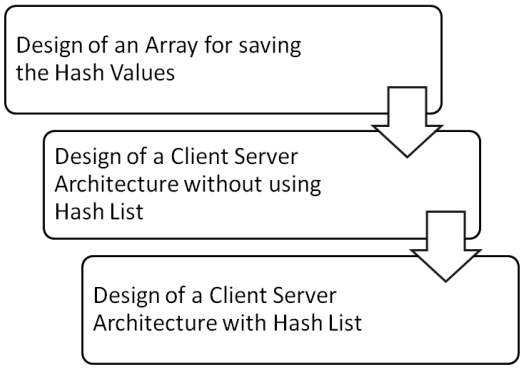

Fig1: Design steps for architecture

The figure 1 shows the designing steps of Architecture development. In these steps, the first step is to save the data in an array form so that the server will recognized the client which requested to access the data. The data saved

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

here is in the form of keys and values. Each key has its own specified value. When the data is requested to access the key is send along with its specified value. The next step in this designing is to design the client server architecture

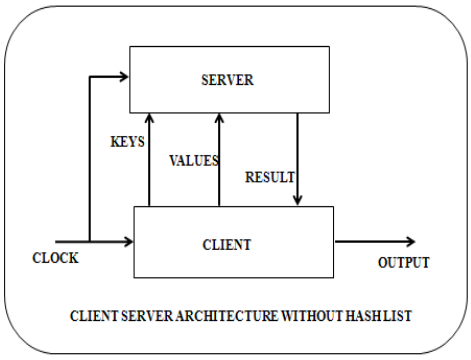

Fig2: Client Server Architecture Design with Key-Value Algorithm

In the implemented work, we first design the client server architecture with the Key-Value Architecture. In this architecture the keys and values should be sent at a time towards the server. This means that if client wants some data from the server then first it will send the key followed by the value. This key-value pair will be detected by the server and the requested data will be searched. After that this information is preceded to the client. The client and server will perform all the operation in same clock cycle. Figure2 shows the basic design of the client server architecture with the key-value algorithm.

This will increase the speed of response of data to be accessed from server to client. The delay in the system is increased as the input came in the serial form i.e. first the key is send to the server then the value is send and after this the output response is generated. This means that the output will get as one output per clock cycle. The key has been send followed by the corresponding value. This will consume more time as the data sending part work on serial mode rather than parallel mode. So we proposed a system architecture which will work in the parallel mode. We are designing a system which will generate all the outputs at a single clock cycle. The outputs came across the single clock cycle.

VII. SIMULATION RESULTS

The simulation results for the proposed architecture are shown in the above figure. For the simulation of work we have used the ModelSim software 6.3f version. This version provides better result in less response time.

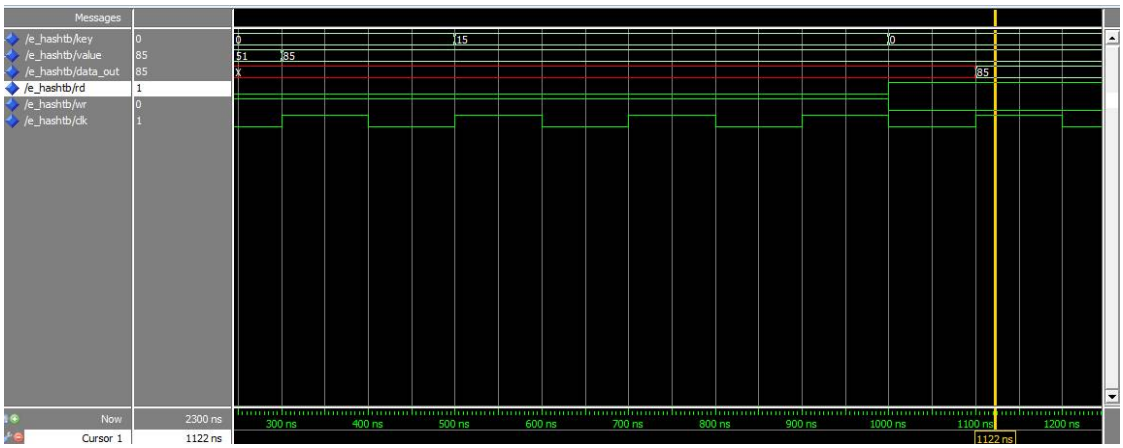

Fig 3: Array output Window

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

In the above figure3, the simulation output is shown. This is the output window for getting the stored key-values from the array. We are designing this array for getting the faster response. All the key-value pair stored in an array are generated at the output as shown in the figure3.

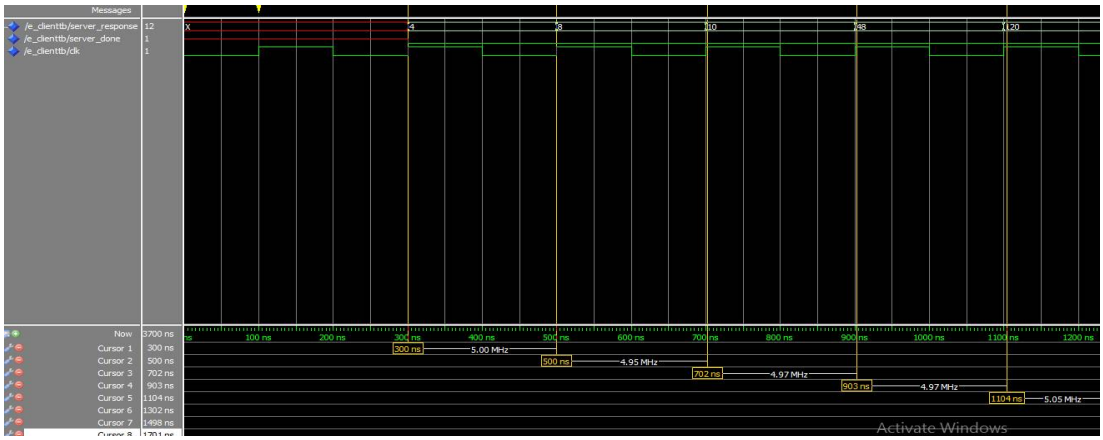

Fig 4: Output of key Value algorithm

The figure 4 above shows the result for client server architecture without using the Hash list architecture. Here one request is send at a single clock cycle. The first output appears at 300ns the second output appears at the 500ns and so on. There is a delay of 200ns in each output response i.e. the number of clock cycles required for respond is more. This means if we want to send 8 bit data we required 8 clock cycles. So to improve this disadvantage we proposed the new architecture named as Hash List based server architecture.

VIII. CONCLUSION AND FUTURE WORK

The Proposed work will result in increase in server request per second. It also increases efficiency of the server to respond to the request. This architecture can improve the performance as the existing one, as it requires the system to send and receive requests continuously. In this 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.

As now we are currently working on a GET request based model for data passing between client and server. In future, we can use the POST, DELETE and other models which are more complex but are secure as compared to the GET model.

REFERENCES

[1] Sheng Li, Hyeontaek Lim, Victor W. Lee, Jung Ho Ahn, AnujKalia, Michael Kaminsky, David G. Andersen, Seongil O, Sukhan Lee, PradeepDubey Intel Labs, "Architecting to Achieve a Billion Requests per Second Throughput on a Sin-gle Key-Value Store Server Platform" IEEE, Carnegie Mellon University, Seoul National University, 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 Plat-form"IEEE-2010
[5] René riessl,MeysamPeykanu, Jens Hagemeyer, Mario Porrmann, Stefan Krupop, Michavordem Berge, Thomas Kie-sel 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. Palecz-ny, "Workload analysis of a large-scale key-value store" in SIGMETRICS, 2012.
[7] A. Belay, G. Prekas, A. Klimovic, S. Grossman, C. Kozyra-kis, 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 AS-PLOS, 2014.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

[10] R. Kapoor, G. Porter, M. Tewari, G. M. Voelker, and A. Vah-dat, "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 USE-NIX 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 Parameterized FPGA-Based Skeleton for Pairwise Biological Sequence Alignment", FCCM"07, pp.275-278, 2007.