



(12) 发明专利

(10) 授权公告号 CN 112637080 B

(45) 授权公告日 2022.11.01

(21) 申请号 202011466483.6

(22) 申请日 2020.12.14

(65) 同一申请的已公布的文献号  
申请公布号 CN 112637080 A

(43) 申请公布日 2021.04.09

(73) 专利权人 中国科学院声学研究所  
地址 100190 北京市海淀区北四环西路21号  
专利权人 中科海网(苏州)网络科技有限公司

(72) 发明人 宋曼谷 沙猛 郭志川 王可

(74) 专利代理机构 北京方安思达知识产权代理有限公司 11472  
专利代理师 杨青 李彪

(51) Int.Cl.

H04L 47/125 (2022.01)

H04L 69/22 (2022.01)

H04L 49/90 (2022.01)

H04L 49/901 (2022.01)

G06F 13/28 (2006.01)

(56) 对比文件

US 2008262984 A1, 2008.10.23

CN 110297785 A, 2019.10.01

CN 109902043 A, 2019.06.18

CN 210112025 U, 2020.02.21

审查员 邹海芳

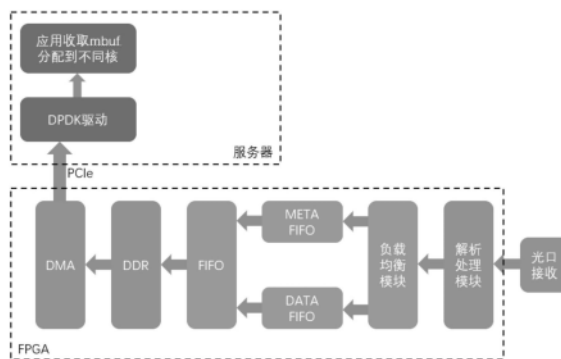
权利要求书1页 说明书4页 附图1页

(54) 发明名称

一种基于FPGA的负载均衡处理系统

(57) 摘要

本发明的公开了一种基于FPGA的负载均衡系统,该系统包括:FPGA加速卡和服务器,FPGA加速卡包括:解析处理模块、负载均衡模块、FIFO缓存模块、DDR缓存和DMA模块;服务器上设置DPDK驱动模块;解析处理模块用于对数据链路层进行解析;负载均衡模块用于提取数据包的五元组信息,根据负载均衡算法对数据包进行计算,确定该数据包所属的队列;将每个数据包和每个数据包的负载均衡结果分别发送至FIFO缓存模块;FIFO缓存模块用于将数据包负载均衡结果插入到数据包以太网前导码之后,并对合并后的数据包进行缓存;DDR模块用于缓存FIFO缓存模块通过AXI总线传输的数据,通过DMA方式传输到服务器;DPDK驱动模块用于根据每个数据包所携带的负载均衡结果,将该数据包拷贝到相应队列中。



1. 一种基于FPGA的负载均衡系统,用于采集和处理高速网络数据包;其特征在于,该系统包括:FPGA加速卡和服务器,两者通过PCIe接口通信;其中,FPGA加速卡包括:解析处理模块、负载均衡模块、FIFO缓存模块、DDR模块和DMA模块;服务器上设置DPDK驱动模块;

所述解析处理模块,用于对数据链路层进行解析,将解析得到的数据包发送到负载均衡模块;

所述负载均衡模块,用于提取数据包的五元组信息,根据负载均衡算法对数据包进行计算,确定该数据包所属的队列;将每个数据包和每个数据包的负载均衡结果分别发送至FIFO缓存模块;

所述FIFO缓存模块包括:META FIFO、DATA FIFO和FIFO;

META FIFO,用于缓存数据包的负载均衡结果;

DATA FIFO,用于缓存数据包;

FIFO,用于将数据包的负载均衡结果插入到数据包以太网前导码之后,将数据包的数据放在负载均衡结果之后,将合并后的数据包通过AXI总线传输到DDR模块;

所述DDR模块,用于缓存FIFO缓存模块通过AXI总线传输的数据;

所述DMA模块,用于将DDR的缓存数据通过DMA方式传输到服务器;

所述DPDK驱动模块,用于根据每个数据包所携带的负载均衡结果,将该数据包拷贝到相应队列中。

2. 根据权利要求1所述的基于FPGA的负载均衡系统,其特征在于,所述负载均衡模块的具体实现过程为:

提取数据包的五元组信息:源IP地址,目的IP地址,源端口,目的端口和协议号;

计算五元组CRC32的校验值,将源IP地址作为CRC32输入计算校验值,将此校验值作为初始向量,同时把源端口作为输入计算第一校验值;

将目的IP地址作为CRC32输入计算校验值,将此校验值作为初始向量,同时把目的端口作为输入计算第二校验值;

将协议号作为CRC32输入计算第三校验值;

对三个校验值进行异或运算,对要分配的核数取模获取该数据包的负载均衡结果;

将该数据包和该数据包的负载均衡结果分别发送至FIFO缓存模块。

3. 根据权利要求1所述的基于FPGA的负载均衡系统,其特征在于,所述FIFO缓存模块对一个数据包的具体处理过程为:

当META FIFO不为空时,将以太网前导码放入到FIFO中,然后读取META FIFO中的负载均衡结果,将一个数据包的负载均衡结果放入到FIFO;然后读取DATA FIFO中的该数据包的数据放入到FIFO中。

4. 根据权利要求3所述的基于FPGA的负载均衡系统,其特征在于,所述DPDK驱动模块的具体实现过程为:

从服务器内存中读取数据包;

当识别到以太网数据帧前导码则认为收取到新数据包;首先将位于前导码后面的负载均衡结果字段放入到mbuf->RSS字段,然后将位于负载均衡结果后面的数据包负载部分放入mbuf->data字段;

上层应用收取数据包后,根据mbuf->RSS字段将mbuf分配到不同的队列上。

## 一种基于FPGA的负载均衡处理系统

### 技术领域

[0001] 本发明涉及网络通信技术领域,特别涉及一种基于FPGA的负载均衡处理系统。

### 背景技术

[0002] 网络通信的迅速发展使得网络流量和网络速率不断提升,对网络数据进行高速的采集和处理已经成为学术界和工业界的重点。CPU单核对高速网络数据包处理能力有限,需要前置网络在前端分流。例如全流量留存系统采集网络中的数据包,之后对数据包进行解析。全流量留存系统需要使用负载均衡将数据包分配到不同核心上面,同时需要将相同业务的数据包尽量分配到同一个核心上,以减少不同处理器核心切换带来的开销。全流量留存系统使用传统网卡提供的RSS算法来实现负载均衡,但此算法对数据包进行负载均衡后,相同业务的数据包往往不能分配到同一个核心上,而网卡使用的RSS算法又无法对其进行修改;全流量留存系统也可以使用软件的方法实现负载均衡,以实现将相同业务分配到同一核心的需求,但软件实现负载均衡性能不理想,不能达到线速的要求,延迟较高,无法满足数据包实时处理的需求。

[0003] FPGA经过快速发展,已经成为数据包处理的主流平台之一。使用FPGA进行数据包处理具有高吞吐率和低延时的优点,因而被应用于网络数据处理的各个领域。在高速网络数据处理的过程中,首先使用FPGA对网络数据包进行处理,再将处理后的数据传输至上位机。在负载均衡方面,首先利用FPGA对数据包进行负载均衡,将负载均衡的结果和数据包通过DMA一同发送至上位机,上位机通过驱动程序进行接收分配和处理。利用DPDK驱动程序可以高效地实现对数据包的这些操作。

[0004] DPDK是一组快速处理数据包的开发平台及接口,它通过轮询的方式处理数据包。它向应用层提供了简单易行且高效的数据包处理方式,使得网络开发更加方便,可以说DPDK正成为网络数据处理的一种标准。通过DPDK驱动可以将FPGA采集的数据包放入到mbuf中,同时可以将此数据包的负载均衡的结果放入mbuf->RSS字段,以方便在后续处理中将数据包分配到不同的处理器核心上。

### 发明内容

[0005] 本发明针对全流量留存系统负载均衡的需求,提出了一种低复杂度的负载均衡系统,通过对现有的负载均衡方案的改进和完善,能够实现零丢包的负载均衡,同时引入的延迟低,对全流量留存系统所做的软件修改小。

[0006] 为实现上述目的,本发明提出了一种基于FPGA的负载均衡系统,用于采集和处理高速网络数据包;该系统包括:FPGA加速卡和服务器,两者通过PCIe接口通信;其中,FPGA加速卡包括:解析处理模块、负载均衡模块、FIFO缓存模块、DDR缓存和DMA模块;服务器上设置DPDK驱动模块;

[0007] 所述解析处理模块,用于对数据链路层进行解析,将解析得到的数据包发送到负载均衡模块;

[0008] 所述负载均衡模块,用于提取数据包的五元组信息,根据负载均衡算法对数据包进行计算,确定该数据包所属的队列;将每个数据包和每个数据包的负载均衡结果分别发送至FIFO缓存模块;

[0009] 所述FIFO缓存模块,用于将数据包负载均衡结果插入到数据包以太网前导码之后,并对合并后的数据包进行缓存;

[0010] 所述DDR模块,用于缓存FIFO缓存模块通过AXI总线传输的数据;

[0011] 所述DMA模块,用于将DDR的缓存数据通过DMA方式传输到服务器;

[0012] 所述DPDK驱动模块,用于根据每个数据包所携带的负载均衡结果,将该数据包拷贝到相应队列中。

[0013] 作为上述系统的一种改进,所述负载均衡模块的具体实现过程为:

[0014] 提取数据包的五元组信息:源IP地址,目的IP地址,源端口,目的端口和协议号;

[0015] 计算五元组CRC32的校验值,将源IP地址作为CRC32输入计算校验值,将此校验值作为初始向量,同时把源端口作为输入计算第一校验值;

[0016] 将目的IP地址作为CRC32输入计算校验值,将此校验值作为初始向量,同时把目的端口作为输入计算第二校验值;

[0017] 将协议号作为CRC32输入计算第三校验值;

[0018] 对三个校验值进行异或运算,对要分配的核数取模获取该数据包的负载均衡结果;

[0019] 将该数据包和该数据包的负载均衡结果分别发送至FIFO缓存模块。

[0020] 作为上述系统的一种改进,所述FIFO缓存模块包括:META FIFO、DATA FIFO和FIFO;

[0021] META FIFO,用于缓存数据包的负载均衡结果;

[0022] DATA FIFO,用于缓存数据包;

[0023] FIFO,用于将数据包的负载均衡结果插入到数据包以太网前导码之后,将数据包的数据放在负载均衡结果之后,将合并后的数据包通过AXI总线传输到DDR模块。

[0024] 作为上述系统的一种改进,所述FIFO缓存模块对一个数据包的具体处理过程为:

[0025] 当META FIFO不为空时,将以太网前导码放入到FIFO中,然后读取META FIFO中的负载均衡结果,将一个数据包的负载均衡结果放入到FIFO;然后读取DATA FIFO中的该数据包的数据放入到FIFO中。

[0026] 作为上述系统的一种改进,所述DPDK驱动模块的具体实现过程为:

[0027] 从服务器内存中读取数据包;

[0028] 当识别到以太网数据帧前导码则认为收取到新数据包;首先将位于前导码后面的负载均衡结果字段放入到mbuf->RSS字段,然后将位于负载均衡结果后面的数据包负载部分放入mbuf->data字段;

[0029] 上层应用收取数据包后,根据mbuf->RSS字段将mbuf分配到不同的队列上。

[0030] 与现有技术相比,本发明具有的优势为:

[0031] 1、相比于传统网卡使用的RSS负载均衡算法,本发明的系统使用数据包的五元组信息来进行负载均衡,相同业务的数据包会被分配到同一个处理器核心上,关联性强同时具有较好的均衡效果;

- [0032] 2、本发明的系统具有低复杂度、低延时、高吞吐率和资源占用率少优点；
- [0033] 3、本发明的负载均衡方法适用于数据中心高速网络数据包处理、网络虚拟化NFV、SDN、全流量留存等相关领域。

### 附图说明

- [0034] 图1是本发明的基于FPGA的负载均衡处理系统总体框架图；
- [0035] 图2是本发明的负载均衡算法流程图；
- [0036] 图3是本发明的数据包合成的示意图。

### 具体实施方式

[0037] 下面通过附图对本发明的技术方案做进一步详细描述。

[0038] 高速网络数据包处理需要使用负载均衡方法将数据包分配到不同的处理器核心上进行解析,同时要求相同业务的数据包能够分配到同一个核心上。使用传统网卡提供的负载均衡算法及软件方法难以实现这些需求。

[0039] 现有使用FPGA对以太网数据应用负载均衡算法,根据负载均衡算法的结果,将具有相同结果的数据包(即之后要分配到同一路的数据包)放入到同一个FPGA的硬件FIFO中。如果需要将数据包分配到N路,则在FPGA中必须对应建立N个硬件FIFO来缓存数据包。而本发明将负载均衡的结果插入到以太网前导码后,由于每个数据包前导码后带有负载均衡结果标签,只需要将负载均衡后的数据包放入1个FPGA的硬件FIFO中,不需要使用多个FPGA硬件FIFO分别缓存具有不同负载均衡结果的数据包。因此,本发明的系统使用的FPGA硬件资源要少,实现的FPGA负载均衡方法复杂度低。

[0040] 如图1所示,本发明提出了一种基于FPGA的负载均衡处理系统,利用FPGA改进和实现现有的负载均衡算法,可以实现零丢包的负载均衡,且引入的延迟较小。本发明的系统包括FPGA加速卡和服务器,两者之间通过PCIe接口来进行数据传输。

[0041] FPGA加速卡包括:解析处理模块、负载均衡模块、FIFO缓存模块、DDR和DMA模块;服务器上设置DPDK驱动模块;

[0042] 解析处理模块处理光口收到数据包,然后将数据包送入负载均衡模块;负载均衡模块通过算法得出数据包应该被分配到哪个队列;FIFO缓存模块将结果放入到META FIFO中,将数据包放入到DATA FIFO中,然后将META FIFO和DATA FIFO中的内容合并放入到PKT FIFO中,以将负载均衡的结果放入到以太网前导码后方;最后将PKT FIFO的数据通过AXI总线传输到DDR上;DDR的缓存数据通过DMA模块传输到服务器。

[0043] 服务器通过DPDK驱动模块收取数据包,当驱动识别到以太网前导码时,将位于以太网前导码后面的负载均衡信息填入到mbuf->RSS字段,将数据负载填入mbuf->data。之后根据mbuf->RSS字段将数据包分配到不同的核上面。即DPDK驱动模块根据数据包中所携带的负载均衡的标签结果,快速将数据包分配到相应的队列中。

[0044] 负载均衡模块的实施步骤为:

[0045] 提取数据包的五元组信息(源IP地址,目的IP地址,源端口,目的端口,协议号)计算CRC32校验值;根据要均衡的CPU核数对上面得出的CRC32校验值进行模运算;输出负载均衡结果。

[0046] FIFO缓存模块的实施步骤为：

[0047] 将负载均衡模块输出的结果放入META FIFO中，将数据包放入到DATA FIFO中。然后将META FIFO中数据包1的负载均衡结果放入到PKT FIFO中，再将DATA FIFO中数据包1负载放入到PKT FIFO中，如此依次交替进行。

[0048] 该系统的具体处理过程包括：

[0049] 首先，高速光口接收数据包，通过物理层解析后，数据流以标准的XGMII接口形式输出，通过解析处理模块进行数据链路层解析，输出以太网数据流及控制信号。

[0050] 接着以太网数据流和控制信号进入到负载均衡模块。在负载均衡模块中，对数据包应用负载均衡算法，具体的处理流程如图2。首先，根据以太网数据帧标准从数据包的对应位置提取五元组信息；然后，对上述五元组中的源IP、源端口、目的IP、目的端口、协议号进行CRC32处理。具体处理过程如下：

[0051] 1、在第一次执行CRC32运算时，先确定其初始向量为0x1edc6f41，然后把32bit的源IP信息作为CRC32的输入并得出其校验结果；

[0052] 2、对于1中得到的校验值，把其作为新的初始向量对源端口信息进行校验运算并得到相应的结果；

[0053] 3、对于目的信息，同样先选定CRC32的初始向量，同时把目的IP信息输入到CRC32运算模块中，得到运算结果；

[0054] 4、把上述的结果作为CRC32运算模块的初始向量对目的端口信息进行校验运算；

[0055] 5、对于协议信息，同样进行如上所述的操作，选定初始向量并求校验值；

[0056] 6、把上述源信息、目的信息及协议信息对应的三个校验值进行异或运算；

[0057] 7、对上述的异或结果进行取模运算，得出负载均衡结果（即需要将此数据包分到哪一路）作为元数据。

[0058] FIFO缓存模块负责将负载均衡结果放入到META FIFO中，将数据帧放入到DATA FIFO中。FIFO缓存模块对META FIFO和DATA FIFO的具体处理流程如图3。首先当META FIFO不为空时，将以太网前导码放入到PKT FIFO中，然后读取META FIFO中的数据，将其中数据包1的元数据放入到PKT FIFO；然后读取DATA FIFO中的数据放入到PKT FIFO中；接着将以太网前导码放入到PKT FIFO中，读取META FIFO中数据包2的元数据和DATA FIFO中数据包2的数据，依次放入PKT FIFO中；如此交替进行。

[0059] PKT FIFO中的数据会通过AXI总线写入DDR，并经过DMA上传至服务器内存。

[0060] 服务器通过DPDK驱动收取数据包。驱动从主机内存中识别到以太网数据帧前导码则认为收取到了新数据包。驱动首先将位于前导码后面的负载均衡结果字段放入到mbuf->RSS字段，然后将位于负载均衡结果后面的数据包负载部分放入mbuf->data字段。上层应用收取数据包后，根据mbuf->RSS字段将mbuf分配到不同的核上。

[0061] 最后所应说明的是，以上实施例仅用以说明本发明的技术方案而非限制。尽管参照实施例对本发明进行了详细说明，本领域的普通技术人员应当理解，对本发明的技术方案进行修改或者等同替换，都不脱离本发明技术方案的精神和范围，其均应涵盖在本发明的权利要求范围当中。

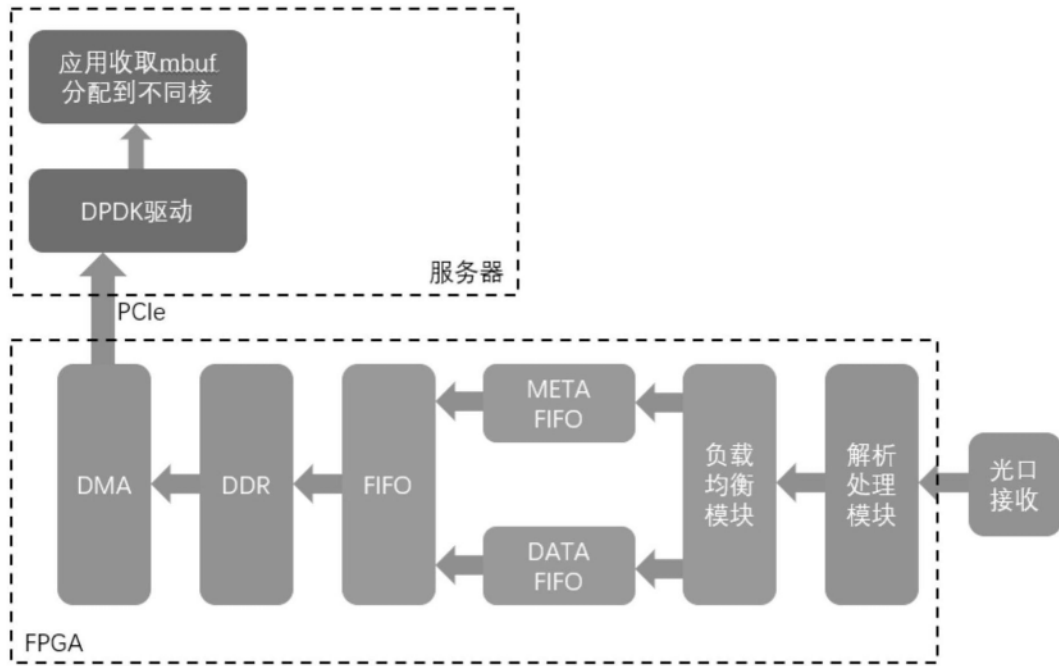


图1

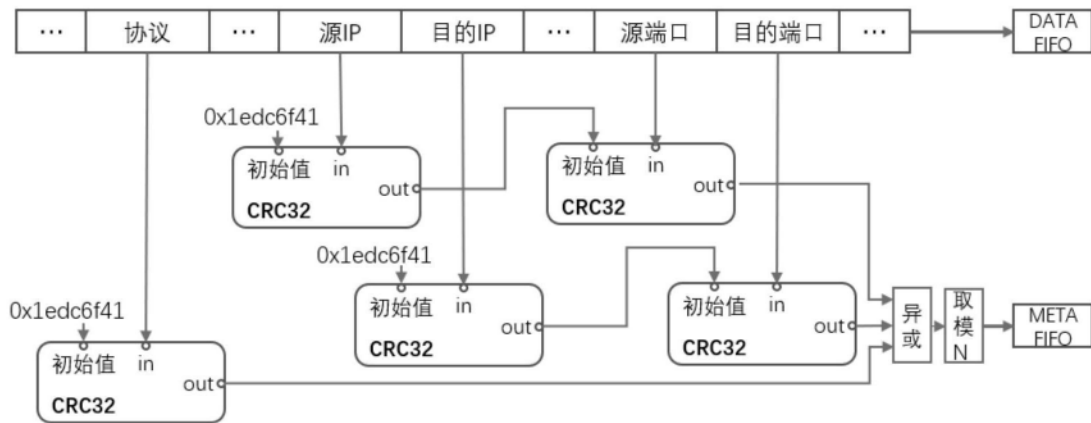


图2

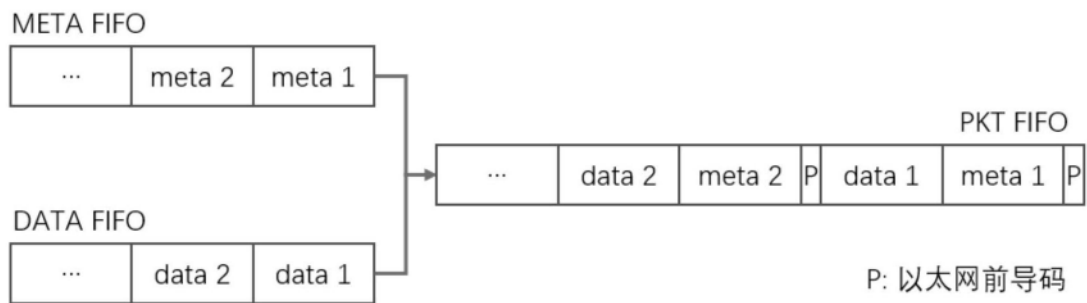


图3