

一种基于 FPGA 的卷积神经网络加速器设计与实现

仇 越¹, 马文涛², 柴志雷^{1,2}

(1 江南大学 物联网工程学院, 江苏 无锡 214122;

2 数学工程与先进计算国家重点实验室, 江苏 无锡 214125)

摘 要: 针对卷积神经网络模型 ZynqNet 现有 FPGA 实现版本中卷积运算单元并行度低, 存储结构过度依赖片外存储等问题, 提出一种针对 ZynqNet 的 FPGA 优化设计. 设计了双缓冲结构将中间运算结果放到片内以减少片外存储访问; 将数据位宽从 32 位降为 16 位; 设计了具有 64 个卷积运算单元的并行结构. 实验结果表明, 在 ImageNet 测试准确度相同的情况下, 本文所提出的设计工作频率可达 200 MHz, 运算速率峰值达到 1.85 GMAC/s, 是原 ZynqNet 实现的 10 倍, 相比 i5-5200U CPU 可实现 20 倍加速. 同时, 其计算能效达到了 NVIDIA GTX 970GPU 的 5.4 倍.

关键词: 卷积神经网络; 现场可编程门阵列(FPGA); ZynqNet; 并行计算; 加速

中图分类号: TP39

文献标识码: A

文章编号: 1000—7180(2018)08—0068—05

Design and Implementation of a Convolutional Neural Network Accelerator Based on FPGA

QIU Yue¹, MA Wen-tao², CHAI Zhi-lei^{1,2}

(1 School of Internet of Things, Jiangnan University, Wuxi 214122, China;

2 State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214125, China)

Abstract: In the hardware design of ZynqNet implemented on FPGA, the parallelism of convolution unit is low and the storage structure is almost dependent on off-chip memory. A FPGA accelerator optimization is proposed based on ZynqNet and it is easy to apply in other CNN models. The double buffering stores intermediate result of the network into the chip to reduce off-chip access; The data precision is changed from 32 bits to 16 bits, thus a parallel structure of 64 convolution operation units is designed to improve computing parallelism. The ImageNet results show that the optimized accelerator based on FPGA can achieve peak performance of 1.85 GMAC/s under 200 MHz, it is 10 times speedup compared to the original ZynqNet and 20 times speedup compared to i5-5200U CPU. In terms of performance power ratio, the FPGA accelerator is 5.4 times of NVIDIA GTX 970GPU version.

Key words: convolutional Neural Network (CNN); field-programmable gate Array(FPGA); ZynqNet; parallelism computing; acceleration

1 引言

随着深度学习技术在众多领域的广泛使用, 作为其基础模型的卷积神经网络(Convolutional Neural Network, CNN)也越来越受重视, 并被广泛应用于诸多领域^[1-2].

目前基于 FPGA 的卷积神经网络加速器已成为业界热点, 此前研究者在 FPGA 上实现的 CNN 多是 10 层以内的小规模网络^[3-4], 输入图像尺寸在 128 像素×128 像素以下, 功能多为简单的图像分类或识别. 苏黎世联邦理工学院团队虽然基于 SqueezeNet 模型^[5] 训练出了适合在 FPGA 上运行

的 ZynqNet 网络,并针对网络实现了 FPGA 硬件加速,但在设计方案中卷积运算单元并行度较低,存储结构过度依赖片外存储,导致计算速度对比 CPU 没有明显优势。

本文根据 ZynqNet 的网络特点及硬件方案中存在的问题设计了一种高度并行的 FPGA 硬件架构和卷积运算单元,与已有的 ZynqNet 实现及 CPU、GPU 实现进行了性能及结果分析。

2 卷积神经网络和 ZynqNet

2.1 卷积神经网络

CNN 作为一种多层神经网络,典型的网络结构如图 1 所示。通常图像作为原始输入,经过交替出现的卷积层提取特征和池化层子抽样后,得到多张特征图,在网络的最后通过全连接层运算得出结果。

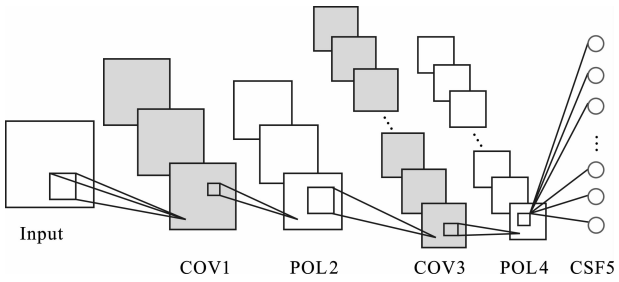


图 1 CNN 网络结构示意图

卷积层通过各种可训练的卷积核对输入数据进行滤波,提取图像特征。卷积计算具体过程如下:取特征图中与卷积核相同大小的区域和卷积核做对应点的乘法,结果求和再加上偏置系数,经激励函数处理后得到一个值,滑动卷积核不断进行卷积操作便生成了特征图。常用激励函数有 ReLU、Sigmoid、Tanh 等,研究表明 ReLU 函数比 Sigmoid 函数和 Tanh 函数有更快的收敛速度^[6]。

ReLU: $y = \max(0, x)$ (1)

池化层对卷积层的输出做子抽样操作,使网络对图像的局部变换有一定自适应性,常用的池化有平均池化和最大池化。平均池化是对 $n \times n$ 的抽样窗口取平均值,如式(2)所示。

$$x_j^l = f(\frac{1}{n} \sum_{i \in M_j} x_i^{l-1} + b_j^l)$$
 (2)

式中, b_j^l 是第 l 层中第 j 个特征图对应的偏置系数; $f(x)$ 为激励函数,常用 ReLU 函数。

全连接层通常在网络的最后几层,这一层中每一个神经元与上一层的神经元全部相连,这种全连

接使得权值数量非常多,因此目前多用全局平均池化代替全连接,达到减少权值,提升运算速度的效果。

2.2 ZynqNet 网络及加速器

近年来涌现出许多优秀的 CNN 网络模型,诸如 AlexNet、GoogleNet、Inception V3 等,这些模型为了提高准确度,网络规模越做越大,然而 2016 年发布的 SqueezeNet 模型,分类准确度接近 AlexNet 但权值规模缩小 50 倍。

为了减少网络权值个数,ZynqNet 同样使用了 SqueezeNet 定义的 Fire 模块,如图 2 所示。Fire 模块由 squeeze 层和 expand 层组成,首先特征图经过 squeeze 层 1×1 的卷积核卷积,然后生成的特征图送入 expand 层经 1×1 和 3×3 的卷积核分别计算,最终结果合并便是输出。但不同的是 ZynqNet 将部分 squeeze 层 1×1 的卷积核改为了 3×3 大小,这样就可以在网络没有池化层的情况下,通过 3×3 的卷积将特征图尺寸缩小。

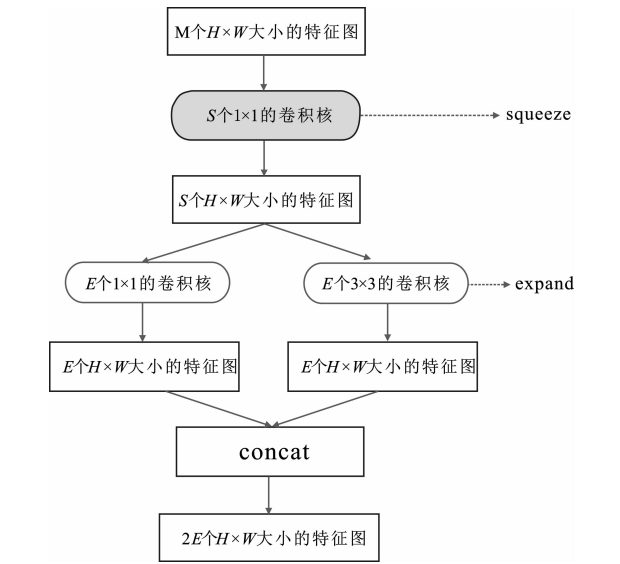


图 2 Fire 模块示意图

ZynqNet 具体的网络结构如表 1 所示。输入层为 ImageNet 数据集预处理后的 3 通道 256×256 大小的图像,首先通过 conv1 层 64 个 3×3 卷积核的计算以及 ReLU 函数激活得到了 64 张 128×128 的输出,这一层计算量为 27 M。然后上一层的结果经由 8 个 Fire 模块和 conv10 的计算输出为 1 024 张 8×8 的特征图,最后经全局平均池化得到了 1 024 个分类,将分类传输到 Softmax 分类器得出最终的图像分类结果。

表 1 ZynqNet 网络结构

层名	输出数据	$s_1 \times 1$ /步长	$s_3 \times 3$ /步长	$e_1 \times 1$ 和 $e_3 \times 3$ /步长	计算量
input	$256 \times 256 \times 3$	—	—	—	—
conv1	$128 \times 128 \times 64$	—	3×3 卷积核/2	—	27 M
fire2	$64 \times 64 \times 128$	—	$16/2$	$64/1$	76 M
fire3	$64 \times 64 \times 128$	$16/1$	—	$64/1$	48 M
fire4	$32 \times 32 \times 256$	—	$32/2$	$128/1$	76 M
fire5	$32 \times 32 \times 256$	$32/1$	—	$128/1$	48 M
fire6	$16 \times 16 \times 512$	—	$64/2$	$256/1$	76 M
fire7	$16 \times 16 \times 384$	$64/1$	—	$192/1$	48 M
fire8	$8 \times 8 \times 512$	—	$112/2$	$256/1$	41 M
fire9	$8 \times 8 \times 736$	$112/1$	—	$368/1$	29 M
conv10	$8 \times 8 \times 1024$	—	1×1 卷积核/1	—	46 M
avgpool10	$1 \times 1 \times 1024$	—	—	—	—

原有 ZynqNet 加速器为了充分利用 FPGA 的硬件并行特点,实现了单个卷积核内部的并行和同一卷积层中多个卷积核的并行.但是在一定程度上限制了加速的性能.另外在存储结构中,原有 ZynqNet 加速器将每一层的中间结果都缓存到片外,但导致硬件设计流水线不能充分运行.同时,在每层计算前和计算后都需要进行大量数据的传输,极大地降低了 FPGA 的运行效率.

3 ZynqNet 的 FPGA 实现方案

3.1 硬件整体架构设计

卷积神经网络层与层之间存在数据依赖,一般很难做到层间并行,但是 CNN 层内包含大量对不同特征图的卷积运算,因此可以实现特征图间并行,卷积核间并行以及卷积核内并行.另外 CNN 各层存在极高的相似性,可以复用单层的运算资源来实现整个网络,因此设计了如图 3 的层串行硬件架构.

首先通过 ARM 端配置相应层的控制寄存器,随后经 AXI 总线从外部存储器 DDR3 中加载图像到图像缓冲区,为了保持整个架构的流水操作,图像缓冲区需要存下两张图像.虽然整个网络的权值过大无法全部存储到 FPGA 片内,但是只要 DDR3 的 IO 带宽足够,就可以实现实时读取权值到权值缓冲区.由于 ZynqNet 对权值带宽的最高要求为 3.6 GB/s,加上输入数据带宽和输出数据带宽,总带宽要求为 4 GB/s,Zynq XC-7Z045 中 DDR3 的 IO 带宽为 12.8 GB/s,因此可以实时读取权值.另外偏移量参数足够小可以直接存储到片内,共占用 16 个 BRAM.

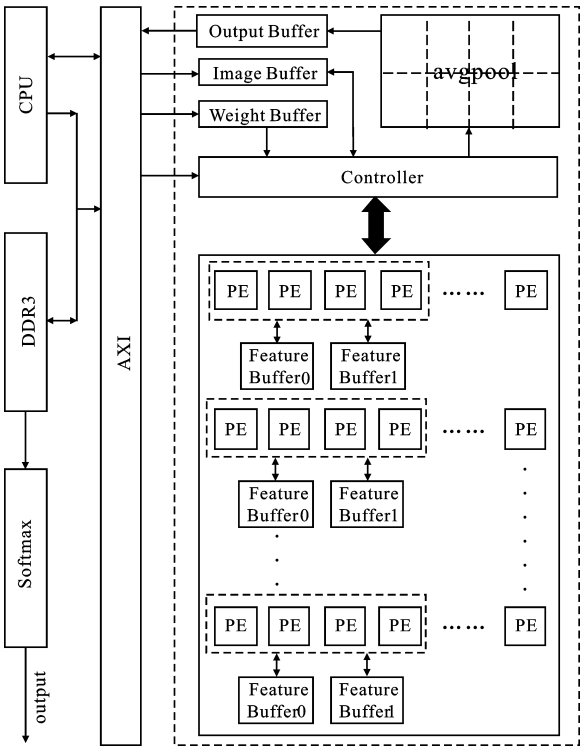


图 3 整体硬件架构图

卷积运算模块是整个硬件架构的核心,由 $N \times M$ 个 PE 单元以及特征图缓冲区组成.因为 Zynq XC-7Z045 中 DSP 资源共计 900 个,每个 PE 负责 3×3 大小的卷积运算,所以最终设计了 64 个 PE,形成 8×8 的并行卷积加速阵列.特征图缓冲区为双缓冲结构,一个是工作缓冲负责存储上一层网络的运算结果,另一个是结果缓冲负责存储本层网络的运算结果.由于 FPGA 片内存储资源有限,因此设计每 4 个 PE 共享 2 个缓冲,共计 32 个特征图缓冲.双缓冲结构使得网络中间运算结果不需要传输到外部存储器,实现了层间数据复用,极大的提升了整个网络的运算速度.然而 ZynqNet 中间运算结果最大达到 2 M,双缓存就是 4 M,Zynq XC-7Z045 片上存储空间只有 2.4 M,根本无法在 FPGA 片内存下.因此只能妥协将 conv1,Fire2,Fire3 这三层的运算结果存储到 DDR3 中,需要时再读回到片上,其它各层运算结果直接存储到特征图缓冲区.

控制模块主要负责运算前的配置,各个模块间的交互和运算过程中的控制.通过控制模块将 conv10 的运算结果从卷积运算模块传输至平均池化模块,平均池化模块包含 8 个池化单元,每个池化单元由三层加法树和数据缓冲构成.经过池化得到 1 024 个分类,将它们存储到输出缓冲区,再经由 AXI 总线传输至 DDR3 中,最后用 Softmax 分类得

出最终结果. 整个硬件架构充分利用了FPGA片上存储的特点,通过双缓冲策略,做到了最大化的数据复用,减少了不必要的片外访存.

3.2 卷积运算单元设计

卷积运算模块作为整个结构的核心部分,每个单元相对独立,结构相同,可充分发挥FPGA的并行计算特性. 具体设计如图4所示,每4个PE共享1个图像寄存器,1个权值寄存器,1个偏移量缓冲和1个PE控制单元,极大地节省了FPGA内部有限的存储资源.

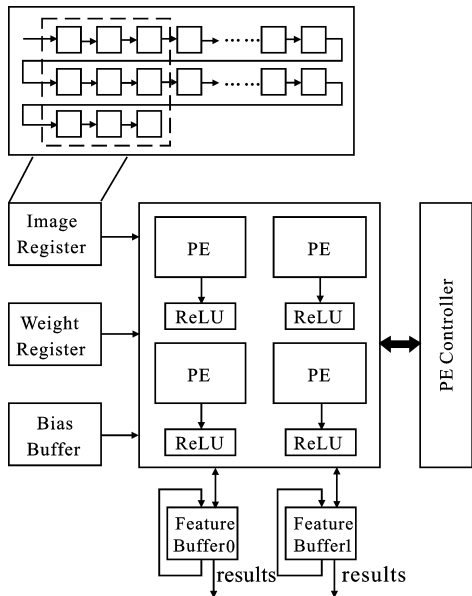


图4 卷积运算单元

图像寄存器采用典型的行缓冲设计^[3],为了支持最大 256×256 的图像输入,行缓冲包括 $256 \times 2 + 3$ 个寄存器,但ZynqNet网络其他层输入图像较小,因此行缓冲中实际存储的数据个数由PE控制单元掌控. 运算时数据通过控制模块串行输入到图像寄存器,输入的同时实现了 3×3 的划窗操作,每次从中选取9个数据分发至4个PE单元参与运算. 由于整个网络的偏移量足够小,可以直接存储到片内,只需在FPGA上电时将偏移量加载到16个偏移量缓冲中,PE就可以实时读取数据,减少外部存储访问.

单个PE负责9个卷积数据,9个网络权值,1个来自特征图缓冲区的累加数据和1个偏移量的计算,因此设计了11个DSP组成的4层乘加树. 第一层为4个乘法器,负责4对卷积数据和权值的乘法计算. 第二层与第一层采用级联方式连接,也为4个乘法器,负责卷积数据和权值相乘的同时

累加第一层的结果. 第三层为1个乘法器和1个加法器,乘法器同样负责一对数据和上一层结果的乘加运算,加法器负责上一层3个结果的加法运算. 第四层为1个加法器,负责偏移量或累加数据与上一层2个结果的加法运算. 最终结果经移位操作调整为16位,数据由ReLU函数激活后送至特征图缓冲区.

为了支持网络的无间断流水运算,特征图缓冲区分为工作缓冲和结果缓冲. 计算第 n 层时,首先从存储着第 $n-1$ 层运算结果的工作缓冲中读取数据,通过控制模块广播给各个PE单元,同时为了防止数据丢失还需将广播数据写回到工作缓冲. 随后PE单元读取数据进行卷积操作,运算结果存储到结果缓冲. 如此调用多轮,直至第 n 层全部计算完毕. 值得注意的是,最后一轮调用时数据无需写回,存储内容为空的工作缓冲便是第 $n+1$ 层计算时的结果缓冲. 双缓冲的结构减少了外部存储的访问,提升了整个网络的运算速度,虽然由于con1, Fire2, Fire3的中间结果过大无法完整存储到特征图缓冲区,但是针对中间输出特征图较小的网络模型,双缓冲更具通用性.

3.3 计算精度优化

ZynqNet网络权值为32位浮点数,在FPGA中需要使用3个DSP资源才能完成一次乘加操作,但是针对16位浮点数只需1个DSP就能实现一次乘加. 并且在FPGA内存储数据,16位浮点数比32位能够减少近一半的BRAM占用. 通过检查每个网络参数,发现将其转化为16位浮点数基本不会损失权值信息,对比实验也证实了降低精度对网络的识别准确度影响极低. 因此本文硬件设计中的图像输入和网络权值全部采用half类型即半浮点数,这样极大地提升了网络运算速度.

4 实验结果与分析

4.1 实验环境

本文进行硬件设计的开发板为Xilinx公司的ZC706,它使用Xilinx Zynq XC-7Z045 FPGA芯片. XC-7Z045芯片包含PS端和PL端,PS采用ARM双核处理器,PL端包含218600LUTS的可编程FPGA阵列. 实验采用的测试集是ImageNet数据集中100张 256×256 大小的图片,每个像素的取值范围是 $0 \sim 225$ 之间的所有整数,但在对图像的预处理过程中,将像素值归一化为 $[-0.5, 0.5]$ 之间的浮点数.

为了对比本文提出的 FPGA 加速器和原有实现,以及 CPU、GPU 实现的性能,实验采用了相同的数据集和网络结构,但是所用的输入图像和网络权值的精度不同,CPU、GPU 和原有 FPGA 实现使用 ZynqNet 提供的 32 位浮点数,本文设计的加速器使用 16 位浮点数. 另外,CPU 采用主频 2.2 GHz 的低功耗 i5-5200U 处理器,GPU 采用 NVIDIA 公司生产的 GTX 970,其存储器带宽为 224 GB/s,存储器容量为 4 G.

4.2 实验结果

200 MHz 工作频率下,本文设计的加速器和原有加速器的 Xilinx Zynq XC-7Z045 FPGA 资源使用情况如表 2 所示. 虽然降低计算精度可以大幅减少 BRAM 和 DSP 的占用,但是节省下的资源用于双缓冲存储和增加 PE 个数,因此本文设计的加速器对比原有实现 BRAM 降低 20%,DSP 无明显变化.

表 2 FPGA 资源使用情况

模块	原有设计		本文设计	
	资源	比例/%	资源	比例/%
LUTS	155 198	70.00	138 112	63.18
FFS	91 997	21.04	184 498	42.20
BRAMS	996	91.38	778	71.38
DSPS	771	85.67	768	85.33

针对 100 张 256×256 大小的图像进行分类的计算速度如表 3 所示,其中 CPU 和 GPU 的实现代码均为 Caffe 库. 可以看出,本文设计的加速器只需 27.2 s 便完成了 100 张图片的分类,每秒计算 3.68 帧,是原有加速器的 10 倍,相比 i5-5200U CPU 可实现 20 倍加速,但相对于 GPU 在性能上没有优势.

表 3 计算速度对比

平台	图像数	时间/s	FPS
i5-5200UCPU	100	551.4	0.18
NVIDIA GTX970 GPU	100	5.24	19.08
原有设计 XC-7Z045 FPGA	100	272.3	0.37
本文设计 XC-7Z045 FPGA	100	27.2	3.68

FPGA 在低功耗方面一直表现卓越,本实验使用功耗仪测量了 FPGA 的功耗情况,CPU 和 GPU 的功耗采用其标称的功耗值. 如表 4 所示,本文提出的 FPGA 加速器其性能功耗比是 CPU 的 55 倍,和 GPU 相比也达到了 5.4 倍.

表 4 功耗对比

平台	功耗/W	性能功耗比
i5-5200UCPU	21	0.008 6
NVIDIA GTX970 GPU	220	0.086 7
原有设计 XC-7Z045 FPGA	7.8	0.047 0
本文设计 XC-7Z045 FPGA	7.8	0.471 3

5 结束语

本文针对 ZynqNet 模型现有的 FPGA 硬件实现,提出了深度流水的加速器优化方案. 通过双缓冲结构将网络中间运算结果放到片内以减少片外存储访问;设计了包含 64 个卷积运算单元的硬件架构,大幅提升网络并行度. 在 ImageNet 测试准确度相同的情况下,本文提出的 FPGA 加速器工作频率可达 200 MHz,运算速率峰值达到 1.85 GMAC/s,是原 ZynqNet 实现的 10 倍,相比 i5-5200U CPU 可实现 20 倍加速. 同时,其计算能效达到了 NVIDIA GTX 970GPU 的 5.4 倍.

为了进一步提高 FPGA 加速方案的性能,未来的工作方向主要集中在优化硬件结构和训练网络权值到 8 位定点数,并研究多个 3×3 卷积处理单元的可重构,实现 5×5 和 7×7 等卷积运算,提升 FPGA 加速硬件架构的通用性.

参考文献:

[1] 余萍, 赵继生. 基于线性叠加特征和 CNNs 的图像分类方法[J]. 微电子学与计算机, 2015,32(10):36-40.

[2] Likamwa R, Hou Y, Gao J, et al. RedEye: analog ConvNet image sensor architecture for continuous mobile vision[J]. Acm Sigarch Computer Architecture News, 2016, 44 (3) :255-266.

[3] Qiu J, Wang J, Yao S, et al. Going deeper with embedded FPGA platform for convolutional neural network[C] // Acm/sigda International Symposium on Field-Programmable Gate Arrays. New York, NY, USA, ACM, 2016:26-35.

[4] Zhang C, Li P, Sun G, et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks[C] // Acm/sigda International Symposium on Field-Programmable Gate Arrays. New York, NY, USA, ACM, 2015:161-170.

[5] Iandola Forrest N, Moskewicz Matthew W, Ashraf K, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size[J]. Computer Vision and Pattern Recognition, arXiv preprint arXiv:1602.07360, 2016.

- [3] 张联镇, 黄传河, 曾毓珑, 等. SDN 中基于时延优化的多控制器部署方案[J]. 计算机工程与设计, 2015, 36(5):1121-1125.
- [4] Murat K, Arjan D. Quality of Service (QoS) in software defined networking (SDN): a survey[J]. Journal of Network & Computer Applications, 2017, 80(15): 200-218.
- [5] Gonzalez A J, Nencioni G, Helvik B E, et al. A fault-tolerant and consistent SDN controller[C]//Global Communications Conference 2016 IEEE. Washington, IEEE, 2016:1-6.
- [6] Athmiya N S, Shobha K R, Sarimela V. Feasibility study and implementation of open flow based SDN controller for tactical scenario[C]//2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). Bangalore; IEEE, 2016:789-794.
- [7] Schiff L, Schmid S, Canini M. Ground control to major faults: towards a fault tolerant and adaptive SDN control network[C]//2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W). Toulouse, IEEE, 2016: 90-96.
- [8] Schmid S, Suomela J. Exploiting locality in distributed SDN control[C]//Acm Sigcomm Workshop on Hot Topics in Software Defined Networking. New York, ACM, 2013:121-126.
- [9] Fonseca P, Bennesby R, Mota E, et al. A replication component for resilient OpenFlow-based networking

- [C]//2012 Network Operations and Management Symposium. Maui, IEEE, 2012:933-939.
- [10] Tam S W, Xi K, Chao H J. Scalability and resilience in data center networks: dynamic flow reroute as an example[C]//2011 The Global Communications Conference, Houston, IEEE. 2011:1-6.
- [11] Ko H, Suh D, Baek H, et al. Optimal placement of service function in service function chaining[C]//2016 Eighth International Conference on Ubiquitous and Future Networks, Vienna, IEEE, 2016:102-105.
- [12] Tootoonchian A, Ganjali Y. HyperFlow: a distributed control plane for OpenFlow[C]//2010 Internet Network Management Conference on Research on Enterprise Networking. San Jose, USENIX Association Berkeley, 2010:3-3.
- [13] Kuroki K, Matsumoto N, Hayashi M. Scalable open flow controller redundancy tackling local and global recoveries[C]//2013 The Fifth International Conference on Advances in Future Internet. Barcelona, IARIA, 2013:61-66.

作者简介:

樊自甫 男, (1977-), 硕士, 教授. 研究方向为下一代网络技术、通信运营管理.

杨先辉(通讯作者) 男, (1990-), 硕士研究生. 研究方向为下一代网络技术、SDN. E-mail: 931425356@qq.com.

姚杰 男, (1994-), 硕士研究生. 研究方向为下一代网络技术、SDN.

(上接第 72 页)

- [6] Dahl G E, Sainath T N, Hinton G E. Improving deep neural networks for LVCSR using rectified linear unit-sand dropout[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, Canada, IEEE, 2013:8609-8613.

作者简介:

仇越 女, (1992-), 硕士研究生. 研究方向为可重构计算. E-mail: 942979548@qq.com.

马文涛 男, (1981-), 工程师. 研究方向为分布式计算.

柴志雷 男, (1975-), 博士, 副教授. 研究方向为嵌入式系统设计技术、FPGA 的可重构计算等.