ARON: Application-Driven Reconfigurable Optical Networking for HPC Data Centers

Guojun Yuan(1,2), Roberto Proietti(3), Xiaoli Liu(1,2), Alberto Castro(3), Dawei Zang(1,2), Ninghui Sun(1), CheYu Liu(3), Zheng Cao(1), and S.J.B. Yoo(3)

(1) Institute of Computing Technology, CAS, Beijing, China, cz@ncic.ac.cn
(2) Institute of Computer and Control Engineering, UCAS, Beijing, China, yuanquojun@ncic.ac.cn
(3) Department of Electrical and Computer Engineering, University of California, Davis, California, USA

Abstract We designed and experimentally demonstrated an application-driven adaptive network that supports fast topology reconfiguration by wavelength routing to match applications’ communication characteristics. Even in a small-scale system, 1.25× performance improvement is achieved by performing network reconfiguration.

Introduction

Typical high-performance computing (HPC) data centers run multiple heterogeneous applications that exhibit various communication patterns among the computing nodes. In order to optimize each application performance, it would be desirable to match the inter-node communication network to the specific application. For instance, under neighborhood traffic pattern, Torus can achieve similar performance, but much lower energy consumption than HyperX topology. However, today’s HPC data centers use a single architecture to serve various applications. Recently, the pod, which may consist of several racks or an entire row of computing nodes, has been used as a building block for the access layer of data centers. Typically, more than 80% applications running in HPC data centers are deployed in no more than one pod scale. Hence, to better support the applications running, it is preferable to dynamically match the pod’s physical topology to various applications’ logical communication topologies. Such flexible physical topology reconfiguration is difficult in electrically interconnected systems with fixed cabling but feasible in multi-wavelength optical networks.

Previous works on optical interconnection networks mainly focused on using high bandwidth and low-loss features of optical communications, while little attention has been paid to fast topology reconfigurations. Ref. [5] introduced relatively slow topology reconfigurations using WSSs and optical MEMSs. To achieve faster and agile re-configurations, this paper investigates an application-driven, optically-reconfigurable architecture by combining wavelength routing in arrayed waveguide grating routers (AWGRs) and nano-second-scale wavelength-tunable transceivers. As Fig. 1 shows, corresponding to the user demands or communication patterns, the full system is divided into multiple application regions (AppRegions), which are reconfigured into proper topologies (such as Dragonfly, Torus, and etc.) by tuning the wavelengths of transceivers via a software-defined control plane.

Physical Architecture of ARON

Fig. 1(right) shows the physical architecture of the proposed Application-Driven Reconfigurable Optical Network (ARON), which consists of a basic sub-network and a reconfigurable network. The basic sub-network is a mesh topology that guarantees the minimum full-system connectivity, while the reconfigurable network is used to perform topology reconfiguration. The proposed application-driven pod architecture is based on arrayed waveguide grating routers (AWGRs). A pod consists of m racks, each one containing n nodes, one AWGR (AWGRr) for the basic sub-network, and one AWGR (AWGR) for the reconfigurable network. Each node contains an r-port router with s fixed-lambda transceivers (TRXs) for the basic sub-network and (r-s) tunable TRXs for the reconfigurable sub-network. The control plane contains an application manager (AM), an

![Fig. 1: (Left) The concept of application region partitioning; (Right) Physical architecture of the Pod.](image-url)
OpenFlow controller (OC) and several OpenFlow agents (OAs) responsible for topology reconfiguration via wavelength tuning. There are also \( n \) inter-rack AWGRs (AWGR\(_1\)). Each AWGR\(_r\) has its channels on a wavelength grid that is interleaved with the channel grid of AWGR\(_i\) (i.e. both grids have a channel spacing of 0.8 nm but they are offset of 0.4 nm – see Fig. 2c). In this way, each node can use the same \( r\)-s tunable TRXs for intra or inter-rack reconfiguration by working on a different grid (each tunable TRX makes use of 1 by 2 couplers and splitters to connect with both levels of AWGRs).

Let us assume 64 as typical value for the AWGR port count. Then, we can conclude that in a pod there will be:

- \( m \times n \) Nodes, each one with \( r \) transceivers, \( s \) of which are at fixed lambda;
- \( n=64/(r-s) \) nodes per rack. If \( (r-s)=2 \), then \( n=32 \);
- \( m=64/(r-s) \) racks with two AWGRs per rack. If \( (r-s)=2, m=32 \), and the number of AWGRs in the first layer is 64;
- a second layer of \( n \) AWGRs (AWGR\(_r\)).

**Experimental Testbed and Results**

Fig. 2(a) shows the experiment testbed with eight nodes divided in two racks. Each node is implemented with a Virtex7 FPGA board. Each node has two wavelength-specific TRXs and two TRXs with fast tunable lasers (actually, only node2 and node3 are equipped with tunable lasers due to the limited amount of components available). There are four 8-port AWGRs for intra-rack communication as well as one 16-port AWGR\(_3\) for inter-rack communication (one single AWGR\(_r\) is enough here since there are only two racks). Fig. 2(c) shows the channel spectrum allocation for AWGR\(_3\) and AWGR\(_{2,2}\). Note that, there is a 0.4 nm offset between the AWGRs’ channel center wavelengths grid. This offset guarantees strong isolation between the two grids. Therefore, a node (e.g. FPGA\(_8\)) can use the same tunable lasers to send data to AWGR\(_3\) or AWGR\(_{2,2}\) by properly choosing on what wavelength grid the laser has to operate. The same principle is valid for AWGR\(_{1,2}\). We measured the BER at the two Node5 receivers connected with AWGR\(_3\) and AWGR\(_{2,2}\) during the reconfiguration from Mesh to HyperX (see Fig. 2(b)). Node2 (whose both tunable lasers 1 and 2 were working on AWGR\(_3\) grid) needs to tune its tunable laser 2 on AWGR\(_{1,2}\) grid in order to establish a direct connection with Node4. At the same time, Node5 has now a free TRX that will be used to establish a direct connection with Node7. In both scenarios (Mesh or HyperX) we reported \( 10^{-12} \) BER @ 10Gb/s for a received power of -16 dBm without any notable power penalty.

Regarding the control plane, we implemented an SDN architecture based on OpenFlow. For the controller we used the Ryu SDN framework. For the OF switches (agents), we deployed eight Open vSwitches, one for each node. We developed an Application Manager based on python for orchestrating the experiment. For each topology reconfiguration, the Application Manager delegates on the controller to update the switches’ flow tables (routing tables), and thus, creating topology on demand. The communication between the Application Manager and the controller is implemented by using the REST API. For each REST API command received by the controller, it sends the corresponding OF messages to the

---

Fig. 2: (a) Experimental setup. Inset shows the details of the tunable transceiver. A tunable laser is externally modulated and power split. One splitter arm connects with AWGR\(_3\) while the other arm connects with AWGR\(_{2,2}\). The RX can receive data from both AWGRs but not at the same time and depending from the established topology. (b) Examples of reconfiguration scenarios. (c) Spectral responses of AWGRs.
We proposed and demonstrated an optical network whose architecture can be dynamically reconfigured via a software-defined control plane to match applications’ communication characteristics. An 8-node experimental testbed implementation demonstrates benefits of reconfigurations in terms of latency and throughput.

Acknowledgements

The work at UC Davis was supported in part by NSFC (No.61572464, No.61331008), by 863 Program (No.2015AA01A301), and by Huawei Technologies (No.61572464). The work at CAS was supported in part under DoD Agreement Number: W911NF-13-1-0090.

References