Practical Challenges of Cloud Administration

Mentor: Michael Ferdman
Jeffrey Q. Xiao
Department of Computer Science

INTRODUCTION

The COMPAS Cloud relies on several open-source technologies to facilitate its operation and administration. OpenStack acts as the control plane of the cloud, orchestrating the provisioning, networking, and lifecycle management of virtual machines and other cloud resources. Ansible automates the integration and management of new bare-metal resources within the cloud. OpenStack runs on a set of controllers which are deployed by Ansible and hosted within virtual machines managed by Proxmox. Ceph creates and supplies distributed storage to OpenStack, supporting virtual machine provisioning. Zabbix monitors system health and triggers alerts when critical thresholds are met.

Because the COMPAS Cloud operates as a distributed system, it must maintain a high level of fault tolerance to prevent isolated failures from cascading across the system. This requires careful modification and optimization of system components to ensure continued functionality in the face of hardware or software failures. Enhancing the network infrastructure—through redundant links and switches—helps mitigate the risk of disconnection or bottlenecks during node failures. Similarly, strengthening the monitoring infrastructure enables administrators to proactively detect and resolve emerging issues before they escalate into major disruptions.

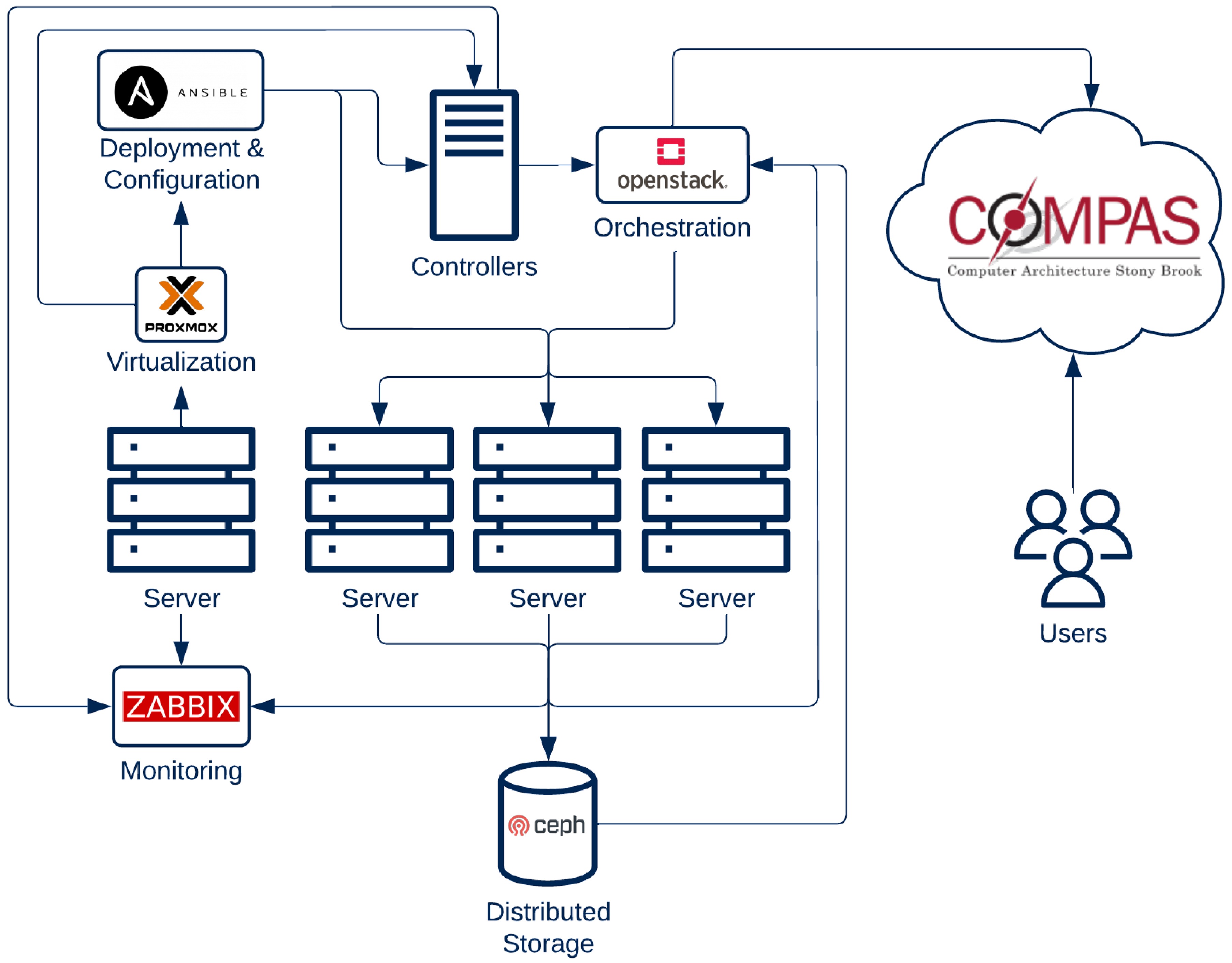

Figure 1. High-Level Architecture of the COMPAS Cloud Infrastructure

METHODS

To improve network resilience and performance, Link Aggregation Control Protocol (LACP) would allow individual nodes to receive traffic across multiple physical interfaces simultaneously. This increases bandwidth while ensuring redundancy in the event of a single interface failure. Additionally, Multi-Chassis Link Aggregation (MLAG) would support active links to two separate core switches, eliminating single points of failure at the switch level and enhancing overall fault tolerance. On the monitoring side, Zabbix configuration modifications to extend automatic discovery rules and data collection parameters would enable administrators to view more granular information about system performance and hardware health.

RESULTS

LACP was successfully implemented across two server racks, encompassing a total of 77 compute nodes. Previously, each node received traffic from two separate 48-port switches, which provided redundancy but limited bandwidth, as only one interface could be active at a time. With LACP, each node was instead connected to a single switch via two aggregated links. Link Aggregation Groups (LAGs) were configured per node, enabling simultaneous use of both interfaces. This change effectively doubled the available network bandwidth per node while preserving interface-level redundancy.

MLAG was not successfully implemented across the two core switches due to compatibility issues with the SONiC operating system and existing network configurations. Ideally, MLAG would allow both switches to act as a single logical unit, enabling seamless failover in the event of a core switch failure. Although LAGs were successfully created and assigned to physical interfaces on both switches, incorrect behavior in SONiC's handling of the Spanning Tree Protocol (STP) prevented the associated interfaces from becoming active.

In Zabbix, modifications to discovery rules and data collection templates enabled agents to gather detailed performance metrics from each host. New item prototypes were developed to track process-level memory usage and CPU utilization. These metrics were paired with custom alert triggers, allowing administrators to detect abnormal behavior in individual processes and intervene before issues escalate.

ACKNOWLEDGEMENT

I would like to thank my advisor, Professor Michael Ferdman, for the opportunity to work on the COMPAS Cloud and gain valuable hands-on experience in cloud administration. I am also deeply grateful to Alan Hau and Hieu Mai for their guidance and support throughout this project—their assistance was instrumental to my progress.

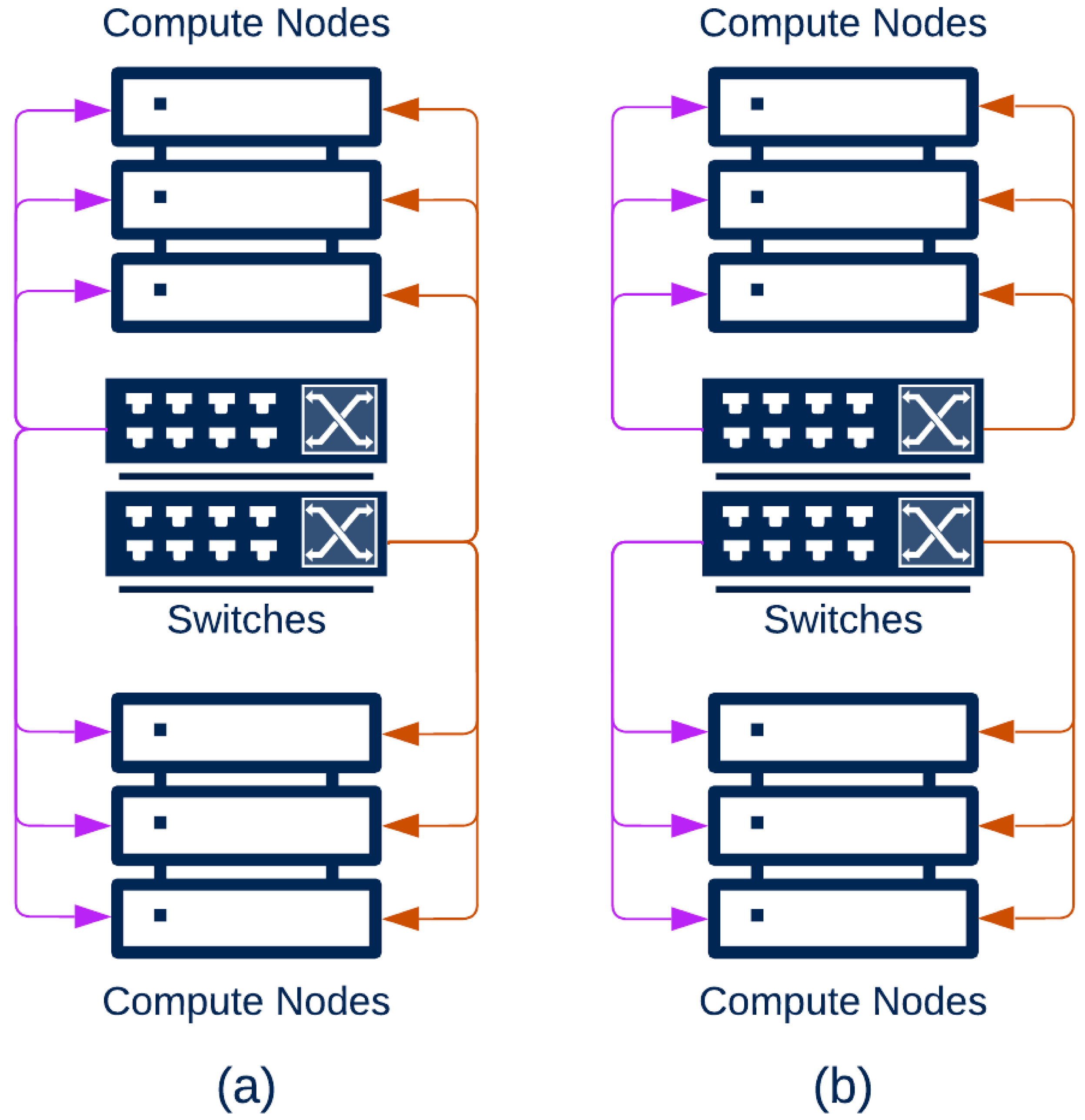

Figure 2. Comparison of Redundant Network Topologies: Independent Dual-NIC Redundancy (a) vs. Link Aggregation (b)

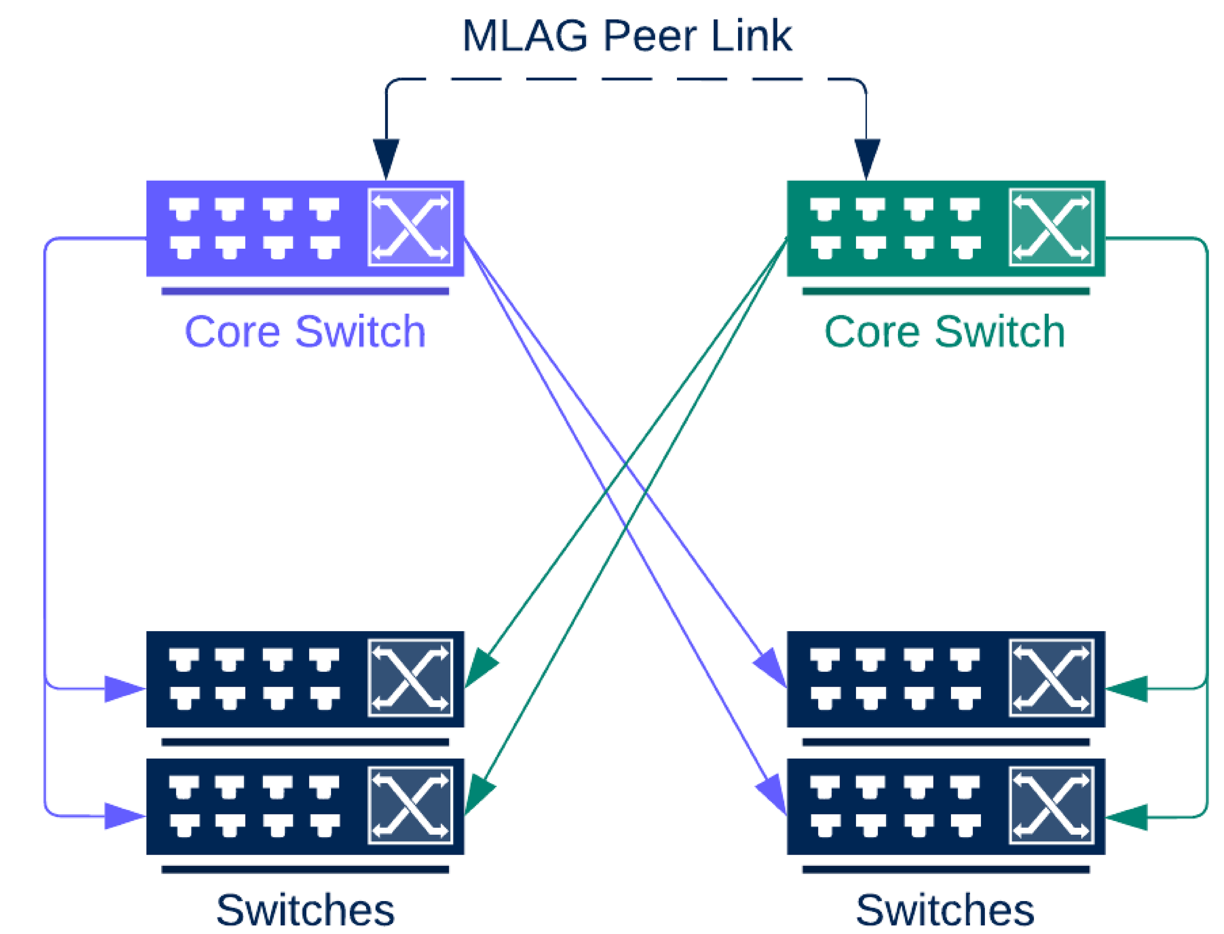

Figure 3. Intended Multi-Chassis Link Aggregation (MLAG) Topology