Cloud-native SmartX Intelligence Cluster for AI-inspired HPC/HPDA Workloads

Jungsu Han
Gwangju Institute of Science and Technology, Rep. of Korea
jshan@nm.gist.ac.kr

Jun-Sik Shin
Gwangju Institute of Science and Technology, Rep. of Korea
jsshin@nm.gist.ac.kr

JinCheol Kwon
Gwangju Institute of Science and Technology, Rep. of Korea
jckwon@nm.gist.ac.kr

JongWon Kim
Gwangju Institute of Science and Technology, Rep. of Korea
jongwon@nm.gist.ac.kr

ABSTRACT

In this Poster, we introduce Cloud-native SmartX Intelligence Cluster for flexibly supporting AI-inspired HPC(high performance computing)/HPDA(high performance data analytics) workloads. This work has been continuously refined from 2013 with a futuristic vision for operating 100 petascale data center. Then, we discuss issues and approaches that come with building Cloud-native SmartX Intelligence Cluster.

CCS CONCEPTS

CCS → General and reference → Cross-computing tools and techniques → Experimentation

KEYWORDS

Cloud native computing, HPC, HPDA, AI, Open Source, Kubernetes

1 INTRODUCTION

To cope with the emerging SDI(Software Defined Infrastructure) paradigm, we have built and operated testbed infrastructure with a concept of SmartX box since 2013. The SmartX Box, which is the hyper-converged whitebox-style resources, should easily offer virtualized/programmable resources by flexibly leveraging various open-source software [1-3].

Meanwhile, we attempt to prepare a futuristic vision for operating 100 petascale AI Cloud data center that supports diversified workloads in car, energy, and healthcare domains. Cloud-native computing is a very agile and flexible approach for a shared cluster to be able to run diversified workloads as containers. Cloud-native computing follows the concept of microservices architecture using open source tools supported by CNCF(Cloud Native Computing Foundation) with the features of CNI(Container Networking Interface) and CSI(Container Storage Interface) on top of Kubernetes orchestration.

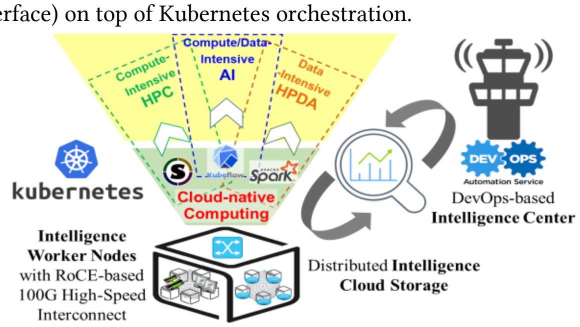

Figure 1: Overall concept of SmartX Intelligence Cluster.

To prepare the transition of a cloud-native paradigm based on our concept of SmartX Box, we design an overall concept of Cloud-native SmartX Intelligence Cluster that empowers CNCF-based open source software solutions for flexibly supporting AI-inspired HPC and HPDA workloads as shown in figure 1.

2 CLOUD-NATIVE SMARTX INTELLIGENCE CLUSTER

2.1 Cloud-native SmartX Intelligence Cluster: Design

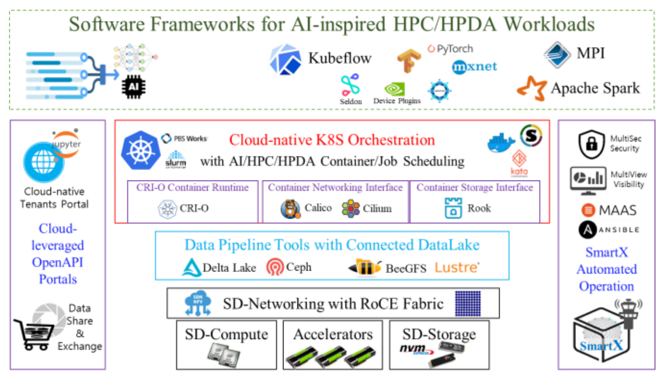

Figure 2: SmartX Intelligence Cluster: Software design.

We design software concept for Cloud-native SmartX Intelligence Cluster as depicted in Figure 2 based on our previous work [4]. We carefully select open-source software solutions that are compatible with Kubernetes.

The overall software architecture of the Cloud-native SmartX Intelligence Cluster is as follows. SDI-ready infrastructure (SD-Compute/Accelerators/SD-Storage with SD-Networking) should be prepared with the Data Pipeline with Connected DataLake. Kubernetes orchestrates software frameworks for HPC/HPDA/AI workloads with DevOps-based automated operation on top of the SDI-ready infrastructure. Cloud-leveraged OpenAPI Portals provide users with API/UI dashboards to control their jobs in the cluster.

2.2 Cloud-native SmartX Intelligence Cluster: Resource Infrastructure

Figure 3 describes the current hardware configuration of the Cloud-native SmartX Intelligence Cluster. In order to verify and refine Cloud-native SmartX Intelligence Cluster, we prepare a mini-scale real testbed. The five Intelligence worker nodes include Nvidia Titan V and Geforce GTX 1080 Ti. The cluster has a total

36TB SATA HDD, 24TB SATA SSD and 15.36TB NF1 NVMe with four intelligence cloud storage nodes. Intelligence Center monitors and controls all nodes by leveraging DevOps automation tools. The nodes are tightly inter-connected through RoCE based 100Gbps-capable high-speed interconnect.

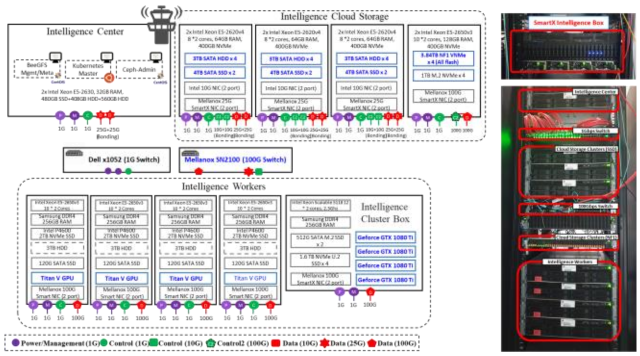

Figure 3: SmartX Intelligence Cluster: Resource infrastructure.

2.3 Cloud-native SmartX Intelligence Cluster: Issues & Approaches

Leveraging open-source software solutions without proper configuration and tuning can degrade the performance. In this section, we discuss issues and our approaches that come with building Cloud-native SmartX Intelligence Cluster.

Physical Inter-connect: Networking can be a performance bottleneck for distributed workloads, so RDMA is a widely used option for HPC due to zero-copy networking. However, for compatibility with non-RDMA nodes, we adopt RDMA over converged Ethernet (RoCE). In addition, RoCE shows 3~30 times low delay than Ethernet from the experiment in our environment.

Kubernetes CNI plugin: Container networking can also become one of serious bottleneck points [5] for the data-intensive workloads. Kubernetes provides various CNI options, and we consider two candidates, Weave and Calico. Weave inter-connects containers via VXLAN-based overlay networking, and VXLAN en(de)capsulation degrades overall networking performance [6]. On the other hand, Calico supports BGP-based container networking without any capsulation overhead. With benchmarking, we verify that Calico shows higher performance than Weave plugin. Thus, we decide to use the Calico BGP plugin that provides BGP mode overlay networking improving network performance with no tunneling.

Kubernetes CSI Plugin: AI-inspired HPC/HPDA workloads require a high-speed parallel file system and data store for huge amounts of data. To select a high-speed parallel file system, we measure performances of several parallel file systems. As a result, we choose BeeGFS [4]. In order to connect BeeGFS to Kubernetes cluster, we should customize configuration since its CSI plugin is not completely developed. For the cloud-based distributed storage, we use Ceph CSI plugin, which is a popular open-source software for distributed storage to provide persistent volume (PV) on Kubernetes.

Container Runtimes for HPC/HPDA/AI: Kubernetes provides several container runtimes. Docker is one of the popular container runtimes, which supports various workloads with software frameworks (Open MPI/Kubeflow/Apache Spark). For HPC intensive workloads, specialized container runtimes such as Singularity and Shifter are introduced due to performance and security issues. However, the current version of Kubernetes does not easily support the specialized container runtimes. In order to deal with this issue, CRI (Container Runtime Interface) has been introduced, complying with the standard of OCI (Open Container Initiative). Although it is still in its infancy, as the latest version of Kubernetes begins to support CRI, we expect to be able to run multiple container runtimes for diversified workloads in near future.

3 CONCLUSION

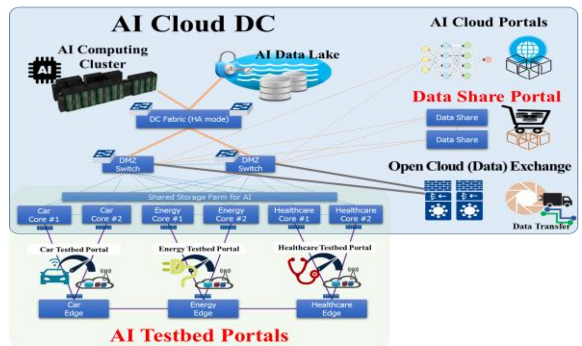

Figure 4: Draft design of AI Cloud data center.

Figure 4 shows the blueprint of the AI Cloud data center, which will be operational in 2023 with national funding support. By leveraging our on-going work on SmartX, we have designed and refined the Cloud-native Intelligence SmartX Cluster to prepare a futuristic vision for the operation of 100 petascale data center.

ACKNOWLEDGMENTS

This work was supported by GIST Research Institute (GRI) grant funded by the GIST in 2019. This work was also supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00421, Cyber Security Defense Cycle Mechanism for New Security Threats).

REFERENCES

[1] J. Kim. "Realizing Diverse Service Over Hyper-converged Boxes with SmartX Automation Framework", *Proc. of Complex, Intelligent, and Software Intensive Systems*, Springer, Cham, 2017.
[2] J. Kim, et al. "OF@TEIN: An OpenFlow-enabled SDN testbed over international SmartX Rack sites", *Proc. of the Asia-Pacific Advanced Network*, 36, 2013.
[3] A.C. Risdianto, J. Shin and J. Kim. "Building and operating distributed SDN-cloud testbed with hyper-convergent SmartX Boxes", *In International Conference on Cloud Computing*, *Springer*, Cham, 2015.
[4] J. Kwon, et al. "Design and Prototyping of Container-enabled Cluster for High Performance Data Analytics", *Proc. of 2019 International Conference on Information Networking (ICOIN)*, IEEE, 2019.
[5] W. Felter, et al. "An updated performance comparison of virtual machines and Linux containers", Proc. Of Performance Analysis of Systems and Software (ISPASS), 2015 IEEE International Symposium On. IEEE, 2015.
[6] K. Suo, et al, "An Analysis and Empirical Study of Container Networks", *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, pp. 189-197. 2018.