Future Generation Computer Systems 67 (2017) 180–193

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Composable architecture for rack scale big data computing

Chung-Sheng Li[a], Hubertus Franke[a], Colin Parris[b], Bulent Abali[a], Mukil Kesavan[c], Victor Chang[d,*]

[a] IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA
[b] GE Global Research Center, One Research Circle, Niskayuna, NY 12309, USA
[c] VMWare, 3401 Hillview Avenue, Palo Alto, CA 9430, USA
[d] IBSS, Xi'an Jiaotong Liverpool University, Suzhou, China

HIGHLIGHTS

- We propose and present composable datacenter, the next-generation foundation to build Cloud and provide Big Data in the Cloud.
- We explain the elements and technical details for composable datacenter.
- We have undertaken MemcacheD, Giraph, and Cassandra experiments on the composable datacenter and discuss each case.
- We demonstrate our work provides added value for the next-generation Big Data in the Cloud.

ARTICLE INFO

Article history:
Received 31 July 2015
Received in revised form
5 July 2016
Accepted 19 July 2016
Available online 31 August 2016

Keywords:
Big data platforms
Composable system architecture
Disaggregated datacenter architecture
Composable datacenter
Software defined environments
Software defined networking

ABSTRACT

The rapid growth of cloud computing, both in terms of the spectrum and volume of cloud workloads, necessitates re-visiting the traditional rack-mountable servers based datacenter design. Next generation datacenters need to offer enhanced support for: (i) fast changing system configuration requirements due to workload constraints, (ii) timely adoption of emerging hardware technologies, and (iii) maximal sharing of systems and subsystems in order to lower costs. Disaggregated datacenters, constructed as a collection of individual resources such as CPU, memory, disks etc., and composed into workload execution units on demand, are an interesting new trend that can address the above challenges. In this paper, we demonstrate the feasibility of composable systems through building a rack scale composable system prototype using PCIe switches. Through empirical approaches, we develop an assessment of the opportunities and challenges for leveraging the composable architecture for rack scale cloud datacenters with a focus on big data and NoSQL workloads. In particular, we compare and contrast the programming models that can be used to access the composable resources, and develop the implications for the network and resource provisioning and management for rack scale architecture.

1. Introduction

Cloud computing is quickly becoming the fastest growing platform for deploying enterprise, social, mobile, and big data analytic workloads [1–3]. Recently, the need for increased agility and flexibility has accelerated the introduction of software defined environments (which include software defined networking, storage, and compute) where the control and management planes of these resources are decoupled from the data planes so that they are no longer vertically integrated as in traditional compute, storage or switch systems and can be deployed anywhere within a datacenter [4].

The emerging datacenter scale computing, especially when deploying big data applications with large volume (in petabytes or exabytes), high velocity (less than hundreds of microsecond latency), wide variety of modalities (structure, semi-structured, and non-structured data) involving NoSQL, MapReduce, Spark/Hadoop in a cloud environment are facing the following challenges: fast changing system configuration requirements due to highly dynamic workload constraints, varying innovation cycles of system hardware components, and the need for maximal sharing of systems and subsystems resources [5–7]. These challenges are further elaborated below.

[*] Corresponding author.
E-mail addresses: csli@ieee.org (C.-S. Li), frankeh@us.ibm.com (H. Franke), colin.parris@ge.com (C. Parris), abali@us.ibm.com (B. Abali), mukilk@gmail.com (M. Kesavan), ic.victor.chang@gmail.com (V. Chang).

http://dx.doi.org/10.1016/j.future.2016.07.014
0167-739X/© 2016 Elsevier B.V. All rights reserved.

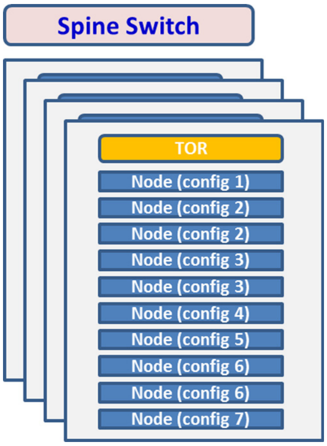

Fig. 1. Traditional datacenter with servers and storage interconnected by datacenter networks.

Systems in a cloud computing environment often have to be configured differently in response to different workload requirements. A traditional datacenter, as shown in Fig. 1, includes servers and storage interconnected by datacenter networks. Nodes in a rack are interconnected by a top-of-rack (TOR) switch, corresponding to the leaf switch in a spine–leaf model. TORs are then interconnected by the Spine switches. Each of the nodes in a rack may have different CPU, memory, I/O and accelerator configurations. Several configuration choices exist in supporting different workload resource requirements optimally in terms of performance and costs. A typical server system configured with only CPU and memory while keeping the storage subsystem (which also includes the storage controller and storage cache) remote is likely to be applicable to workloads which do not require large I/O bandwidth and will only need to use the storage occasionally. This configuration is usually inexpensive and versatile. However, its large variation in sustainable bandwidth and latency for accessing data (through bandwidth limited packet switches) make it unlikely to perform well for most of the big data workloads when large I/O bandwidth or small latency for accessing data becomes pertinent. Alternatively, the server can be configured with large amount of local memory, SSD, and storage. Repeating this configuration for a substantial portion of the datacenter, however, is likely to become very expensive. Furthermore, resource fragmentation arises for CPU, memory, or I/O intensive big data workloads as these workloads often consume one or more dimensions of the resources in its entirety while other dimensions remain underutilized (as shown in Fig. 2), where there exists unused memory for CPU intensive workloads (Fig. 2(b)) or unused CPU for memory intensive workloads (Fig. 2(c)). In summary, no single system configuration is likely to offer both performance and cost advantages across a wide spectrum of big data workload.

Traditional systems also impose identical lifecycles for every hardware component inside the system. As a result, all of the components within a system (whether it is a server, storage, or switches) is replaced or upgraded at the same time. The "synchronous" nature of replacing the whole system at the same time prevents earlier adoption of newer technology at the component level, whether it is memory, SSD, GPU, or FPGA. The average replacement cycle of CPUs is approximately 3–4 years, HDDs and fans are around 5 years, battery backup (i.e. UPS), RAM, and power supply are around 6 years. Other components in a data center typically have a lifetime of 10 years. A traditional system with CPU, memory, GPU, power supply, fan, RAM, HDD, SSD likely has the same lifecycle for everything within the system as replacing these components individually will be uneconomical.

System resources (memory, storage, and accelerators) in traditional systems configured for high throughput or low latency usually cannot be shared across the data center, as these resources are only accessible within the "systems" where they are located. Using the financial industry as an example, they are often required to handle large number of Online Transaction Processing (OLTP) during day time while conducting Online Analytical Processing (OLAP) and business compliance related computation during night time (often referred to as batch window) [8]. OLTP has very stringent throughput, I/O, and resiliency requirements. In contrast, OLAP and compliance workloads may be computationally and memory intensive. As a result, resource utilization could be potentially low if systems are statically configured for individual OLTP and OLAP workloads. Those resources (such as storage) accessible remotely over datacenter networks allow better utilization but the performance in terms of throughput and latency are usually poor, due to a prolonged execution time and constrained quality of service (QoS).

Disaggregated datacenter, constructed as a collection of individual resources such as CPU, memory, HDDs etc., and composed into workload execution units on demand, is an interesting new trend that satisfies several of the above requirements [9]. In this paper, we demonstrate the feasibility of composable systems through building a rack scale composable system prototype using a PCIe switch. Through empirical approaches, we develop an assessment of the opportunities and challenges for leveraging the composable architecture for rack scale cloud datacenters with a focus on big data and NoSQL workloads. We compare and contrast the programming models that can be used to access these composable resources. We also develop the implications and requirements for network and resource provisioning and management. Based on this qualitative assessment and early experimental results, we conclude that a composable rack scale architecture with appropriate programming models and resource provisioning is likely to achieve improved datacenter operating efficiency. This architecture is particularly suitable for heterogeneous and fast evolving workload environments as these environments often have dynamic resource requirements and can benefit from the improved elasticity of the physical resource pooling offered by the composable rack scale architecture.

The rest of the paper is organized as follows: Section 2 describes the architecture of composable systems for a refactored datacenter. Related work in this area is reviewed in Section 3. The software stack for such composable systems is described in Section 4. The network considerations for such composable systems are described in Section 5. A rack scale composable prototype system based on PCIe switch is described in Section 6. We describe the rack scale composable memory in Section 7. Section 8 describes the methodology for distributed resource scheduling. Empirical results from various big data workloads on such systems are reported and discussed in Section 9. Discussions of the implications are summarized in Section 10.

2. Composable system architecture

Composable datacenter architecture, which refactors datacenter into physical resource pools (in terms of compute, memory, I/O, and networking), offers the potential advantage of enabling continuous peak workload performance while minimizing resource fragmentation for fast evolving heterogeneous workloads. Fig. 3 shows rack scale composability, which leverages the fast progress of the networking capabilities, software defined environments, and the increasing demand for high utilization of computing resources in order to achieve maximal efficiency.

On the networking front, the emerging trend is to utilize a high throughput low latency network as the "backplane" of the system.