Big Data Optimization via Next Generation Data Center Architecture

Jian Li

Abstract The use of Big Data underpins critical activities in all sectors of our society. Achieving the full transformative potential of Big Data in this increasingly digital and interconnected world requires both new data analysis algorithms and a new class of systems to handle the dramatic data growth, the demand to integrate structured and unstructured data analytics, and the increasing computing needs of massive-scale analytics. As a result, massive-scale data analytics of all forms have started to operate in data centers (DC) across the world. On the other hand, data center technology has evolved from DC 1.0 (tightly-coupled silos) to DC 2.0 (computer virtualization) in order to enhance data processing capability. In the era of big data, highly diversified analytics applications continue to stress data center capacity. The mounting requirements on throughput, resource utilization, manageability, and energy efficiency demand seamless integration of heterogeneous system resources to adapt to varied big data applications. Unfortunately, DC 2.0 does not suffice in this context. By rethinking of the challenges of big data applications, researchers and engineers at Huawei propose the High Throughput Computing Data Center architecture (HTC-DC) toward the design of DC 3.0. HTC-DC features resource disaggregation via unified interconnection. It offers Peta Byte (PB) level data processing capability, intelligent manageability, high scalability and high energy efficiency, hence a promising candidate for DC 3.0. This chapter discusses the hardware and software features HTC-DC for Big Data optimization.

Keywords Big data · Data center · System architecture

J. Li (✉)
Futurewei Technologies Inc., Huawei Technologies Co. Ltd, 2330 Central Expressway, Santa Clara, CA 95050, USA
e-mail: jian.li1@huawei.com

A. Emrouznejad (ed.), *Big Data Optimization: Recent Developments and Challenges*, Studies in Big Data 18, DOI 10.1007/978-3-319-30265-2_10

1 Introduction

1.1 Challenges of Big Data Processing

During the past few years, applications that are based on big data analysis have emerged, enriching human life with more real-time and intelligent interactions. Such applications have proven themselves to become the next wave of mainstream of online services. At the dawn of the big data era, higher and higher demand on data processing capability has been raised. Given industry trend and being the major facilities to support highly varied big data processing tasks, future data centers (DCs) are expected to meet the following big data requirements (Fig. 1)[1]:

- **PB/s-level data processing capability** ensuring aggregated high-throughput computing, storage and networking;
- **Adaptability** to highly-varied run-time resource demands;
- **Continuous availability** providing 24 × 7 large-scaled service coverage, and supporting high-concurrency access;
- **Rapid deployment** allowing quick deployment and resource configuration for emerging applications.

1.2 DC Evolution: Limitations and Strategies

DC technologies in the last decade have been evolved (Fig. 2) from DC 1.0 (with tightly-coupled silos) to current DC 2.0 (with computer virtualization). Although data processing capability of DCs have been significantly enhanced, due to the limitations on throughput, resource utilization, manageability and energy efficiency, current DC 2.0 shows its incompetence to meet the demands of the future:

- **Throughput**: Compared with technological improvement in computational capability of processors, improvement in I/O access performance has long been lagged behind. With the fact that computing within conventional DC architecture largely involves data movement between storage and CPU/memory via I/O ports, it is challenging for current DC architecture to provide PB-level high throughput for big data applications. The problem of I/O gap is resulted from low-speed characteristics of conventional transmission and storage mediums, and also from inefficient architecture design and data access mechanisms.
  To meet the requirement of future high throughput data processing capability, adopting new transmission technology (e.g. optical interconnects) and new storage medium can be feasible solutions. But a more fundamental approach is

[1]This chapter is based on "High Throughput Computing Data Center Architecture—Thinking of Data Center 3.0", white paper, Huawei Technologies Co. Ltd., http://www.huawei.com.

Fig. 1 Needs brought by big data

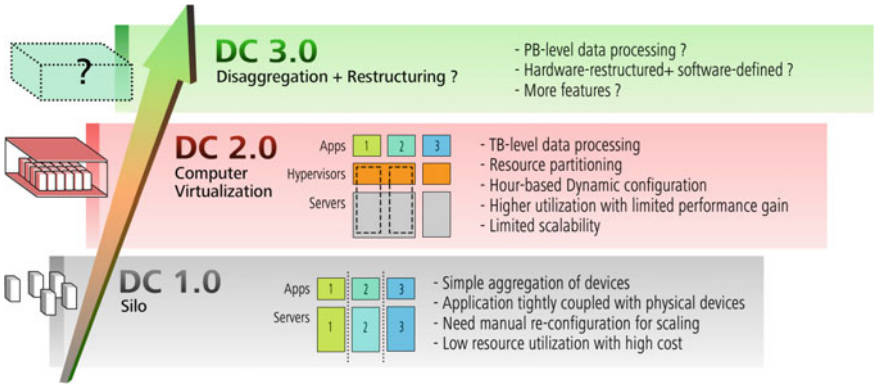

Fig. 2 DC evolution

to re-design DC architecture as well as data access mechanisms for computing. If data access in computing process can avoid using conventional I/O mechanism, but use ultra-high-bandwidth network to serve as the new I/O functionality, DC throughput can be significantly improved.

- **Resource Utilization**: Conventional DCs typically consist of individual servers which are specifically designed for individual applications with various pre-determined combinations of processors, memories and peripherals. Such design makes DC infrastructure very hard to adapt to emergence of various new applications, so computer virtualization technologies are introduced accordingly. Although virtualization in current DCs help improve hardware utilization, it cannot make use of the over-fractionalized resource, and thus making the improvement limited and typically under 30 % [1, 2]. As a cost, high overhead exists with hypervisor which is used as an essential element when implementing computer virtualization. In addition, in current DC architecture, logical pooling of resources is still restricted by the physical coupling of in-rack hardware devices. Thus, current DC with limited resource utilization cannot support big data applications in an effective and economical manner.

One of the keystones to cope with such low utilization problem is to introduce resource disaggregation, i.e., decoupling processor, memory, and I/O from its original arrangements and organizing resources into shared pools. Based on disaggregation, on-demand resource allocation and flexible run-time application deployment can be realized with optimized resource utilization, reducing Total Cost of Operation (TCO) of infrastructure.

- **Manageability**: Conventional DCs only provide limited dynamic management for application deployment, configuration and run-time resource allocation. When scaling is needed in large-scaled DCs, lots of complex operations still need to be completed manually.
  To avoid complex manual re-structuring and re-configuration, intelligent self-management with higher level of automation is needed in future DC. Furthermore, to speed up the application deployment, software defined approaches to monitor and allocate resources with higher flexibility and adaptability is needed.
- **Energy Efficiency**: Nowadays DCs collectively consume about 1.3 % of all global power supply [3]. As workload of big data drastically grows, future DCs will become extremely power-hungry. Energy has become a top-line operational expense, making energy efficiency become a critical issue in green DC design. However, the current DC architecture fails to achieve high energy efficiency, with the fact that a large portion of energy is consumed for cooling other than for IT devices.
  With deep insight into the composition of DC power consumption (Fig. 3), design of each part in a DC can be more energy-efficient. To identify and eliminate inefficiencies and then radically cut energy costs, energy-saving design of DC should be top-to-bottom, not only at the system level but also at the level of individual components, servers and applications.

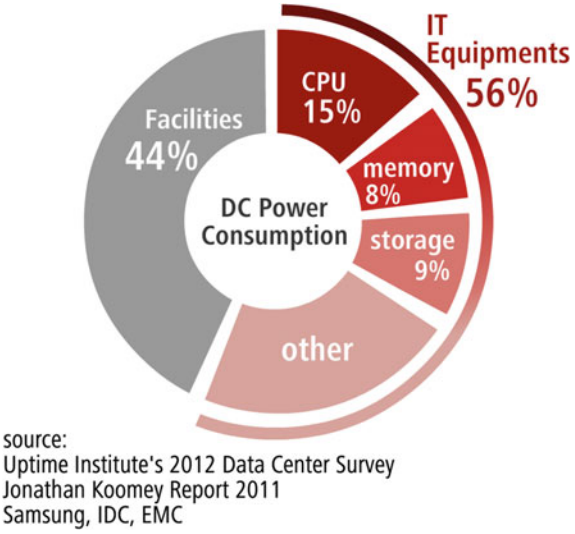

Fig. 3 DC power consumption

1.3 Vision on Future DC

Future DCs should be enabled with the following features to support future big data applications:

- **Big-Data-Oriented**: Different from conventional computing-centric DCs, data-centric should be the key design concept of DC 3.0. Big data analysis based applications have highly varied characteristics, based on which DC 3.0 should provide optimized mechanisms for rapid transmission, highly concurrent processing of massive data, and also for application-diversified acceleration.
- **Adaptation for Task Variation**: Big data analysis brings a booming of new applications, raising different resource demands that vary with time. In addition, applications have different need for resource usage priority. To meet such demand variation with high adaptability and efficiency, disaggregation of hardware devices to eliminate the in-rack coupling can be a key stone. Such a method enables flexible run-time configuration on resource allocation, ensuring the satisfactory of varied resource demand of different applications.
- **Intelligent Management**: DC 3.0 involves massive hardware resource and high density run-time computation, requiring higher intelligent management with less need for manual operations. Application deployment and resource partitioning/allocation, even system diagnosis need to be conducted in automated approaches based on run-time monitoring and self-learning. Further, Service Level Agreement (SLA) guaranteeing in complex DC computing also requires a low-overhead run-time self-manageable solution.
- **High Scalability**: Big data applications require high throughput low-latency data access within DCs. At the same time, extremely high concentration of data will be brought into DC facilities, driving DCs to grow into super-large-scaled with sufficient processing capability. It is essential to enable DCs to maintain acceptable performance level when ultra-large-scaling is conducted. Therefore, high scalability should be a critical feature that makes a DC design competitive for the big data era.
- **Open, Standard based and Flexible Service Layer**: With the fact that there exists no unified enterprise design for dynamical resource management at different architecture or protocol layers, from IO, storage to UI. Resources cannot be dynamically allocated based on the time and location sensitive characteristics of the application or tenant workloads. Based on the common principles of abstraction and layering, open and standard based service-oriented architecture (SOA) has been proven effective and efficient and has enabled enterprises of all sizes to design and develop enterprise applications that can be easily integrated and orchestrated to match their ever-growing business and continuous process improvement needs, while software defined networking (SDN) has also been proven in helping industry giants such as Google to improve its DC network resource utilization with decoupling of control and data forwarding, and centralized resource optimization and scheduling. To provide competitive big data related service, an open, standard based service layer should be enabled in future

DC to perform application driven optimization and dynamic scheduling of the pooled resources across various platforms.

- **Green**: For future large-scale DC application in a green and environment friendly approach, energy efficient components, architectures and intelligent power management should be included in DC 3.0. The use of new mediums for computing, memory, storage and interconnects with intelligent on-demand power supply based on resource disaggregation help achieving fine-grained energy saving. In addition, essential intelligent energy management strategies should be included: (1) Tracking the operational energy costs associated with individual application-related transactions; (2) Figuring out key factors leading to energy costs and conduct energy-saving scheduling; (3) Tuning energy allocation according to actual demands; (4) Allowing DCs to dynamically adjust the power state of servers, and etc.

2 DC3.0: HTC-DC

2.1 HTC-DC Overview

To meet the demands of high throughput in the big data era, current DC architecture suffers from critical bottlenecks, one of which is the difficulty to bridge the I/O performance gap between processor and memory/peripherals. To overcome such problem and enable DCs with full big-data processing capability, we propose a new high throughput computing DC architecture (HTC-DC), which avoids using conventional I/O mechanism, but uses ultra-high-bandwidth network to serve as the new I/O functionality. HTC-DC integrates newly-designed infrastructures based on resource disaggregation, interface-unified interconnects and a top-to-bottom optimized software stack. Big data oriented computing is supported by series of top-to-bottom accelerated data operations, light weighted management actions and the separation of data and management.

Figure 4 shows the architecture overview of HTC-DC. Hardware resources are organized into different pools, which are links up together via interconnects. Management plane provides DC-level monitoring and coordination via DC Operating System (OS), while business-related data access operations are mainly conducted in data plane. In the management plane, a centralized Resource Management Center (RMC) conducts global resource partitioning/allocation and coordination/scheduling of the related tasks, with intelligent management functionalities such as load balancing, SLA guaranteeing, etc. Light-hypervisor provides abstract of pooled resources, and performs lightweight management that focuses on execution of hardware partitioning and resource allocation but not get involved in data access. Different from conventional hypervisor which includes data access functions in virtualization, light-hypervisor focuses on resource management, reducing complexity and overhead significantly. As a systematical DC 3.0 design, HTC-DC also

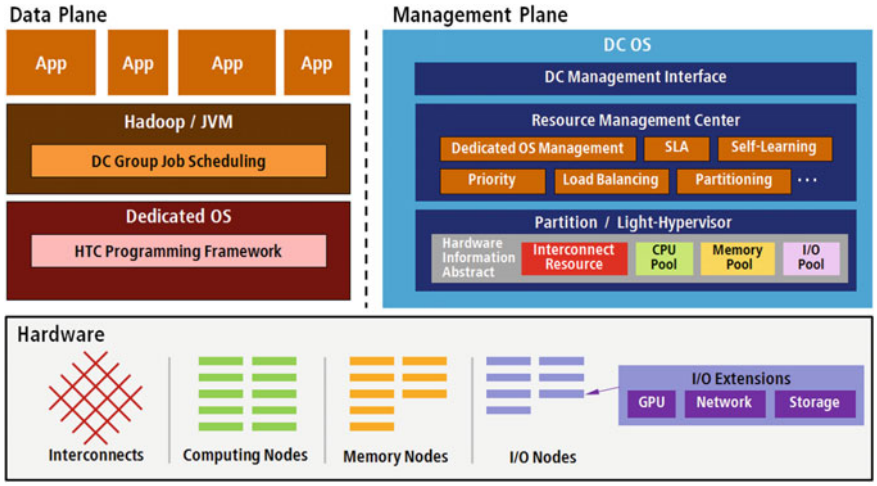

Fig. 4 HTC-DC architecture

provides a complete software stack to support various DC applications. A programming framework with abundant APIs is designed to enable intelligent run-time self-management.

2.2 *Key Features*

Figure 5 illustrates the hardware architecture of HTC-DC, which is based on completely-disaggregated resource pooling. The computing pool is designed with heterogeneity. Each computing node (i.e. a board) carries multiple processors (e.g., x86, Atom, Power and ARM, etc.) for application- diversified data processing. Nodes in memory pool adopt hybrid memory such as DRAM and non-volatile memory (NVM) for optimized high- throughput access. In I/O pool, general-purposed extension (GPU, massive storage, external networking, etc.) can be supported via different types of ports on each I/O node. Each node in the three pools is equipped with a cloud controller which can conduct diversified on-board management for different types of nodes.

2.3 *Pooled Resource Access Protocol (PRAP)*

To form a complete DC, all nodes in the three pools are interconnected via a network based on a new designed Pooled Resource Access Protocol (PRAP). To reduce the complexity of DC computing, HTC-DC introduces PRAP which has

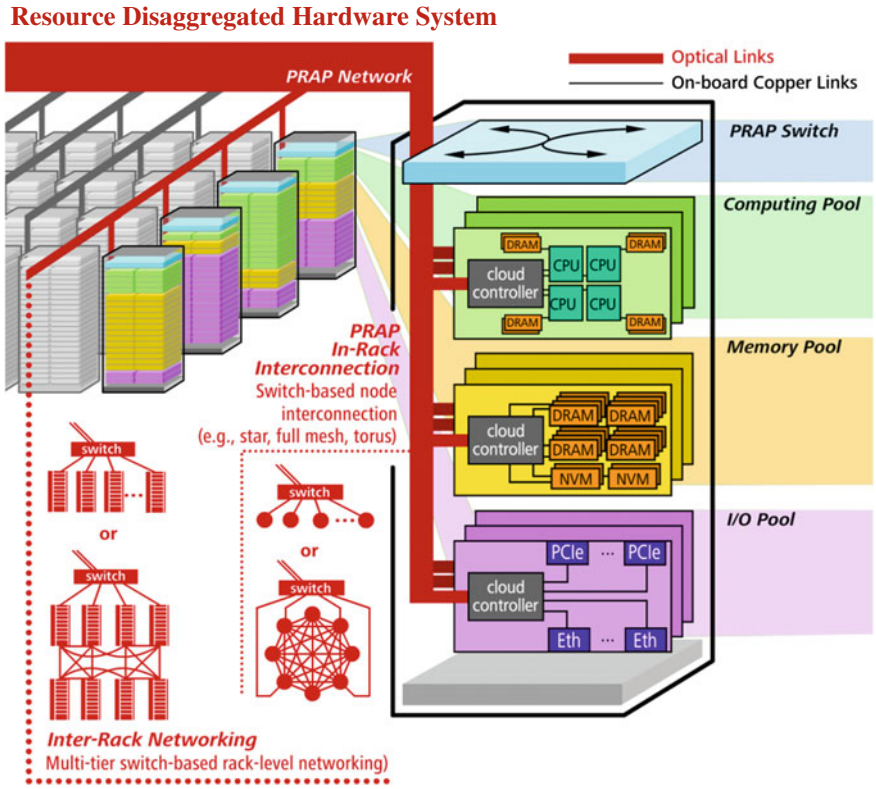

Fig. 5 Hardware architecture of Huawei HTC-DC

low-overhead packet format, RDMA-enabled simplified protocol stack, unifying the different interfaces among processor, memory and I/O. PRAP is implemented in the cloud controller of each node to provide interface-unified interconnects. PRAP supports hybrid flow/packet switching for inter-pool transmission acceleration, with near-to-ns latency. QoS can be guaranteed via run-time bandwidth allocation and priority-based scheduling. With simplified sequencing and data restoring mechanisms, light-weight lossless node-to-node transmission can be achieved.

With resource disaggregation and unified interconnects, on-demand resource allocation can be supported by hardware with fine-granularity, and intelligent management can be conducted to achieve high resource utilization (Fig. 6). RMC in the management plane provides per-minute based monitoring, on-demand coordination and allocation over hardware resources. Required resources from the pools can be appropriately allocated according to the characteristics of applications (e.g. Hadoop). Optimized algorithm assigns and schedules tasks on specific resource partitions where customized OSs are hosted. Thus, accessibility and bandwidth of remote memory and peripherals can be ensured within the partition, and hence end-to-end SLA can be guaranteed. Enabled with self-learning mechanisms,

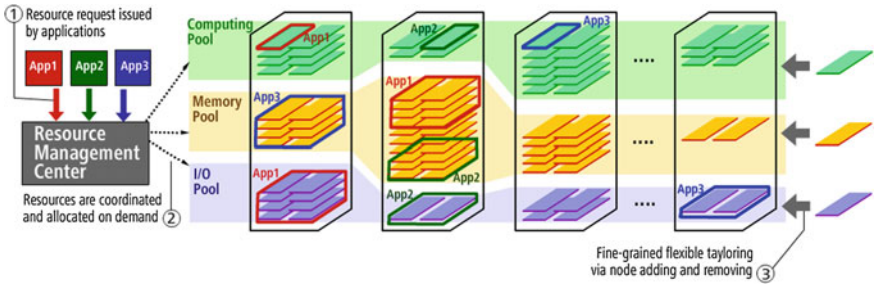

Fig. 6 On-demand resource allocation based on disaggregation

resource allocation and management in HTC-DC requires minimal manual operation, bringing intelligence and efficiency.

2.4 Many-Core Data Processing Unit

To increase computing density, uplift data throughput and reduce communication latency, Data Processing Unit (DPU, Fig. 7) is proposed to adopt lightweight-core based many-core architecture, heterogeneous 3D stacking and Through-Silicon Vias (TSV) technologies. In HTC-DC, DPU can be used as the main computing component. The basic element of DPU is Processor-On-Die (POD), which consists of NoC, embedded NVM, clusters with heavy/light cores, and computing accelerators. With software-defined technologies, DPU supports resource partitioning and QoS-guaranteed local/remote resource sharing that allow application to directly access resources within its assigned partition. With decoupled multi-threading support, DPU executes speculative tasks off the critical path, resulting in enhanced

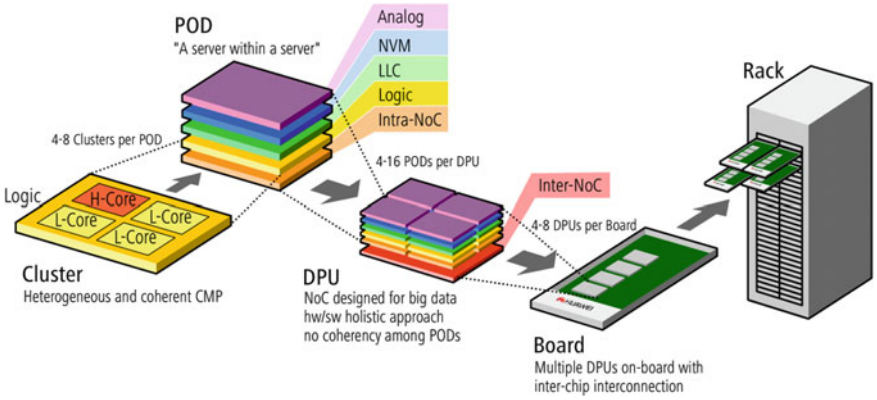

Fig. 7 Many-core processor

overall performance. Therefore static power consumptions can be significantly reduced. Especially, some of the silicon chip area can be saved by using the optimal combinations of the number of synchronization and execution pipelines, while maintaining the same performance.

2.5 NVM Based Storage

Emerging NVM (including MRAM or STT-RAM, RRAM and PCM, etc.) has been demonstrated with superior performance over flash memories. Compared to conventional storage mediums (hard-disk, SSD, etc.), NVM provides more flattened data hierarchy with simplified layers, being essential to provide sufficient I/O bandwidth. In HTC-DC, NVMs are employed both as memory and storage. NVM is a promising candidate for DRAM replacement with competitive performance but lower power consumption. When used as storage, NVM provides 10 times higher IOPS than SSD [4], bringing higher data processing capability with enhanced I/O performance.

Being less hindered by leakage problems with technology scaling and meanwhile having a lower cost of area, NVM is being explored extensively to be the complementary medium for the conventional SDRAM memory, even in L1 caches. Appropriately tuning of selective architecture parameters can reduce the performance penalty introduced by the NVM to extremely tolerable levels while obtaining over 30 % of energy gains [5].

2.6 Optical Interconnects

To meet the demand brought by big data applications, DCs are driven to increase the data rate on links (>10 Gbps) while enlarging the scale of interconnects (>1 m) to host high-density components with low latency. However due to non-linear power consumption and signal attenuation, conventional copper based DC interconnects cannot have competitive performance with optical interconnects on signal integrity, power consumption, form factor and cost [6]. In particular, optical interconnect has the advantage of offering large bandwidth density with low attenuation and crosstalk. Therefore a re-design of DC architecture is needed to fully utilize advantages of optical interconnects. HTC-DC enables high-throughput low-latency transmission with the support of interface-unified optical interconnects. The interconnection network of HTC-DC employs low-cost Tb/s-level throughput optical transceiver and co-packaged ASIC module, with tens of pJ/bit energy consumption and low bit error rate for hundred-meter transmission. In addition, with using intra/inter-chip optical interconnects and balanced space-time-wavelength design, physical layer scalability and the overall power consumption can be enhanced. Using optical transmission that

needs no signal synchronization, PRAP-based interconnects provide higher degree of freedom on topology choosing, and is enabled to host ultra-large-scale nodes.

2.7 DC-Level Efficient Programming Framework

To fully exploit the architectural advantages and provide flexible interface for service layer to facilitate better utilization of underlying hardware resource, HTC-DC provides a new programming framework at DC-level. Such a framework includes abundant APIs, bringing new programming methodologies. Via these APIs, applications can issue requests for hardware resource based on their demands. Through this, optimized OS interactions and self-learning-based run-time resource allocation/scheduling are enabled. In addition, the framework supports automatically moving computing operations to near-data nodes while keeping data transmission locality. DC overhead is minimized by introducing topology-aware resource scheduler and limiting massive data movement within the memory pool.

As a synergistic part of the framework, Domain Specific Language (HDSL) is proposed to reduce the complexity of parallel programming in HTC-DC. HDSL includes a set of optimized data structures with operations (such as Parray, parallel processing of data array) and a parallel processing library. One of the typical applications of HDSL is for graph computing. HDSL can enable efficient programming with demonstrated competitive performance. Automated generation of distributed code is also supported.

3 Optimization of Big Data

Optimizing Big Data workloads differ from workloads typically run on more traditional transactional and data-warehousing systems in fundamental ways. Therefore, a system optimized for Big Data can be expected to differ from these other systems. Rather than only studying the performance of representative computational kernels, and focusing on central-processing-unit performance, practitioners instead focus on the system as a whole. In a nutshell, one should identify the major phases in a typical Big Data workload, and these phases typical apply to the data center in a distributed fashion. Each of these phases should be represented in a distributed Big Data systems benchmark to guide system optimization.

For example, the MapReduce Terasort benchmark is popular a workload that can be a “stress test” for multiple dimensions of system performance. Infrastructure tuning can result in significant performance improvement for such benchmarks. Further improvements are expected as we continue full-stack optimizations on both distributed software and hardware across computation, storage and network layers. Indeed, workloads like Terasort can be very IO (Input-Output) intensive. That said, it requires drastically higher throughput in data centers to achieve better

performance. Therefore, HTC-DC, our high-throughput computing data center architecture works perfectly with such big data workloads.

Finally, we plan to combine this work with a broader perspective on Big Data workloads and suggest a direction for a future benchmark definition effort. A number of methods to further improve system performance look promising.

4 Conclusions

With the increasing growth of data consumption, the age of big data brings new opportunities as well as great challenges for future DCs. DC technology has evolved from DC 1.0 (tightly-coupled server) to DC 2.0 (software virtualization) with enhanced data processing capability. However, the limited I/O throughput, energy inefficiency, low resource utilization and hindered scalability of DC 2.0 have become the bottlenecks to meet the demand of big data applications. As a result, a new, green and intelligent DC 3.0 architecture capable to adapt to diversified resource demands from various big-data applications is in need.

With the design of ultra-high-bandwidth network to serve as the new I/O functionality instead of conventional schemes, HTC-DC is promising to serve as a new generation of DC design for future big data applications. HTC-DC architecture enables high throughput computing in data centers. With its resource disaggregation architecture and unified PRAP network interface, HTC-DC is currently under development to integrate many-core processor, NVM, optical interconnects and DC-level efficient programming framework. Such a DC will ensure PB-level data processing capability, support intelligent management, be easy and efficient to scale, and significantly save energy cost. We believe HTC-DC can be a promising candidate design for future DCs in the Big Data era.

Acknowledgments Many Huawei employees have contributed significantly to this work, among others are, Zhulin (Zane) Wei, Shujie Zhang, Yuangang (Eric) Wang, Qinfen (Jeff) Hao, Guanyu Zhu, Junfeng Zhao, Haibin (Benjamin) Wang, Xi Tan (Jake Tam), Youliang Yan.

References

1. http://www.energystar.gov/index.cfm?c=power_mgt.datacenter_efficiency_consolidation
2. http://www.smartercomputingblog.com/system-optimization/a-data-center-conundrum/
3. http://www.google.com/green/bigpicture/#/datacenters/infographics
4. http://www.samsung.com/global/business/semiconductor/news-events/press-releases/detail?newsId=12961
5. Komalan, M., et.al.: Feasibility exploration of NVM based I-cache through MSHR enhancements. In: Proceeding in DATE'14
6. Silicon Photonics Market & Technologies 2011–2017: Big Investments, Small Business, Yole Development (2012)

Author Biography

Jian Li is a research program director and technology strategy leader at Huawei Technologies. He was the chief architect of Huawei's FusionInsights big data platform and is currently leading strategy and R&D efforts in IT technologies, working with global teams around the world. Before joining Huawei, he was with IBM where he worked on advanced R&D, multi-site product development and global customer engagements on computer systems and analytics solutions with significant revenue growth. A frequent presenter at major industry and academic conferences around the world, he holds over 20 patents and has published over 30 peer-reviewed papers. He earned a Ph. D. in electrical and computer engineering from Cornell University. He also holds an adjunct position at Texas A&M University. In this capacity, he continues to collaborate with leading academic researchers and industry experts.