A survey of cloud resource management for complex engineering applications

Article *in* Frontiers of Computer Science (print) · January 2016

DOI: 10.1007/s11704-015-4207-x

7 authors, including:

Haibao Chen
Huazhong University of Science and Technol…
14 PUBLICATIONS **19** CITATIONS
SEE PROFILE

Song Wu
Huazhong University of Science and Technol…
159 PUBLICATIONS **1,022** CITATIONS
SEE PROFILE

Wenguang Chen
Tsinghua University
93 PUBLICATIONS **770** CITATIONS
SEE PROFILE

Front.Comput.Sci.
DOI

REVIEW ARTICLE

A Survey of Cloud Resource Management for Complex Engineering Applications

Haibao CHEN [1,4], **Song WU** (✉)[1], **Hai JIN** [1], **Wenguang CHEN** [2], **Jidong ZHAI** [2], **Yingwei LUO** [3], **Xiaolin WANG** [3]

1 Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Comput. Sci. and tech., Huazhong University of Science and Technology, Wuhan 430074, China
2 Institute of High Performance Computing, Tsinghua University, Beijing 100084, China
3 Institute of Network Computing and Information Systems, Peking University, Beijing 100871, China
4 School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China

Abstract Traditionally, Complex Engineering Applications (CEAs), which consist of numerous components (software) and require a large amount of computing resources, usually run in dedicated clusters or high performance computing (HPC) centers. Nowadays, Cloud computing system with the ability of providing massive computing resources and customizable execution environment is becoming an attractive option for CEAs. As a new type of Cloud applications, CEA also brings the challenges of dealing with Cloud resources. In this paper, we provide a comprehensive survey of Cloud resource management research for CEAs. The survey puts forward two important questions: 1) what are the main challenges for CEAs to run in Clouds? and 2) what are the prior research topics addressing these challenges? We summarize and highlight the main challenges and prior research topics. Our work can be probably helpful to those scientists and engineers who are interested in running CEAs in Cloud environment.

Keywords Cloud Computing, Complex Engineering Application, Resource Management, Virtualization

1 Introduction

The complexity of product design is pushing the limits of engineering processes and computing, leading to increased demand on computing resources and rapidly escalating costs. How to meet these requirements becomes a challenge that the engineering community must address [1].

In recent years, datacenter-based Cloud computing system is profoundly changing the way people use resources and services.

Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility (like the electricity grid) over a network (typically the Internet). It provides computation, software, and storage services that do not require end-user knowledge of the physical location and system configuration, and offers users a range of benefits including small startup and maintenance costs, economics of scale, customizable execution environment, etc. Because of these benefits, Cloud computing is attracting the attention of more and more Complex Engineering Applications (CEAs) users. However, traditionally, Cloud computing systems have been designed for running web applications, whose resource requirements are different from CEAs. For example, unlike web applications, CEAs typically require low latency and high bandwidth interprocessor communication to achieve best performance. In the case of Cloud, presence of commodity interconnect and effect of virtualization result in interconnect becoming a bottleneck for CEAs. Therefore, it is difficult to achieve anticipated effects when CEAs run directly on current Cloud computing environment [2].

In this paper, we aim to answer two questions from the perspective of Cloud resource management: 1) what are the

Received month dd, yyyy; accepted month dd, yyyy

E-mail: *wusong@hust.edu.cn*

main challenges for CEAs to run in Clouds? and 2) what are the prior research topics addressing these challenges?

The remainder of this paper is organized as follows: In Section 2, we briefly introduce the examples of CEA. In Section 3, we present opportunities for CEA in Cloud computing systems. Next, in Section 4 we propose a taxonomy of the state of the art based on the scientific problems, and review how each problem has been tackled in related research. We provide a discussion on the current challenges and research topics in Section 5. Finally, this paper is concluded in Section 6.

2 What is CEA?

Complex product design is a multidisciplinary optimization process, in which a large number of disciplinary variables are processed to find the optimal or satisfactory solutions. Engineers usually use many software packages to perform scientific computations of aerodynamics, thermodynamics, structure analysis, etc., seeking multidisciplinary design optimization [3]. In order to increase the development efficiency, these multi-disciplinary design, analysis and simulation software packages are generally integrated into a CEA used for complex product design and analysis computations [4]. Although there is no single widely accepted view of CEA in the field, CEA can be presented in many ways [4–6]. In this paper, we refer to CEAs as the engineering applications, the characteristics of which are as follows:

- Many connected components (i.e., software packages with different functions);
- Complex internal executable process, e.g., iterative optimization.
- Diverse and dynamic resources (e.g., CPU, GPU, disk and I/O resources) requirements at different stages during run time;
- Massive demand for computing and I/O resources, etc.

CEAs have been widely applied in the field of industrial manufacturing, e.g., shipbuilding and auto manufacturing, to improve time-intensive design. They often combine complex simulation codes based on fine computational meshes and algorithms for numerical optimization. For example, in the fluid-dynamic design of marine, aeronautical, and automotive transport systems, the shape plays a key role and its detailed analysis often requires the solution of nonlinear partial differential equations (PDE), namely the NavierStokes equations [7], which are particularly expensive from a computational point of view in case of a realistic three dimensional geometry. However, the cost of a simulation, i.e., an objective function evaluation, is CPU time consuming. A typical Ship Design analysis with Navier-Stokes equations takes as much as 10 hours per function evaluation for a three dimensional mesh, for example 2×10^6 nodes [8]. Nonetheless, other applications may require more time (**One characteristic of CEA: massive demand for computing**).

In the rest of this section, we take Ship Design as an example to give a basic picture of CEA. Generally, Ship Design involves hull, machinery, and electrical design. Furthermore, the hull design consists of three parts: overall, structure, and outfitting design, which are interrelated and influenced each other. In this complex system, lots of components or subsystems need to be designed. Besides, the specific problems and contradictions among hull, machinery, and electrical design need to be addressed coordinately.

Because the complexity of Ship Design, it is not possible to finish it in one time, but by a process of successive approximation which is an iterative process of design selection, analysis and validation. Fig. 1 shows the ship design spiral. Each cycle improves the detail level of design, while reducing the number of alternatives. From design assignment to the completion of construction, the process is generally divided into four stages: preliminary design, detail design, production design, and documentation. The successive approximation process design runs through every stage (**One characteristic of CEA: complex internal executable process, e.g., iterative optimization**).

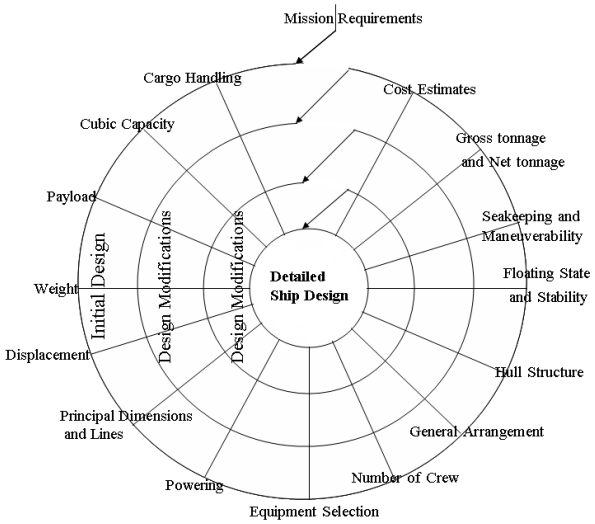


Fig. 1 Procedure of Ship Design.

Contemporary Ship Design relies on computer aided design (CAD) technology involving different computational

analysis software and different running environments. Taking the hydrostatic and hydrodynamic analysis of preliminary design stage as examples, first, the geometric model of the hull is established by modelling software (e.g., TRIBON M3 [9] of Kockums Computer System (KCS) in Sweden). Second, the simulation analysis software (e.g., Fluent [10]) is used to mesh this geometry model and define boundary conditions. Third, the mesh model is invoked with specific parameter values to analyze the hydrodynamic properties of the hull, and the hydrostatic and hydrodynamic resistance is obtained separately after simulated calculation under different conditions. At last, the output parameters such as hull pressure are processed by post-processing software (e.g., ANSYS CFD-Post [11]), and the simulation results are used to provide engineers with guidance for the following design stages. Besides, since a ship is a huge system consisting of millions of parts [12], it will produce large amounts of 3D CAD data and have massive demand on storage and computing during the Ship Design [13] (**Characteristics of CEA: many software packages with different functions and diverse resources requirements at different stages during run time**).

3 Opportunities for Running CEA in Cloud

Compared to traditional computing environments, the main advantages of Cloud computing towards CEAs are as follows.

3.1 Customizable Execution Environment

Different stages and components of CEAs probably exhibit different functional properties and environmental requirements. For example, in the scenario of CEAs having to use legacy programs, different components may be developed with different languages based on different software stacks, even run on different operating systems. Therefore, the complexity of CEAs is largely reflected in their diverse and heterogeneous execution environments, which introduces great challenges to system configuration, management, and resource scheduling of the computing system hosting CEAs.

Compared to traditional computing environments, in addition to offering massive computing and storage resources to CEAs, the biggest advantage of Cloud computing is to provide feasible techniques and efficient solutions to support heterogeneous environments by virtualization technology. Computing environment can be customized according to the specific needs of applications in Cloud computing system. For example, the pre-customized operating system and optimized software stack can be packaged as virtual appliance. Engineers can easily and quickly deploy a customized computing environment for CEAs through adopting suitable virtual appliance, and extend the life-cycle of the legacy applications by shielding heterogeneous hardware resources for upper layer software (operating system, application software, etc.). Another example is that virtualization can greatly improves the configurability of the I/O system of datacenters, and makes the type of file systems and I/O nodes as configurable resources. In fact, this feature is usually missing in traditional computing environment.

3.2 Flexible Resource Management

By means of virtualization technology, Cloud computing can provide new resource management and usage patterns, where virtual machine (VM) is the basic unit of granularity, for CEAs to satisfy their elastic and scalable resource requirements. For one thing, Cloud computing can assign resources according to the characteristics of CEAs. For another thing, Cloud computing can provide standard interfaces for users to customize and optimize the stack software on their VMs.

Different from traditional computing environments, Cloud computing can provide intensive computing resources, heterogeneous execution environments (e.g., different kinds of guest operating systems), and on-demand dynamic resource allocation for CEAs. Besides, taking Electronic Design Automation (EDA) as an example, the Cloud-based EDA not only makes automatic design faster by the virtue of massive parallel computing power and abundant resources supply of Cloud platform, but also allows users to enjoy pay-per-use EDA service [14] which is especially helpful to small-to-medium-sized enterprises/businesses.

3.3 Cost-effectiveness

Cloud computing is probably the most cost efficient method to use, maintain and upgrade. Rather than purchasing expensive equipment, users can reduce their costs by using the resources of Cloud computing service providers. Traditional desktop software costs companies a lot in terms of finance. Adding up the licensing fees for multiple users can prove to be very expensive for the establishment concerned. The Cloud, on the other hand, is available at much cheaper rates and hence, can significantly lower the users' IT expenses. Besides, there are many one-time-payment, pay-as-you-go and other scalable options available, which makes it very reasonable for the users.

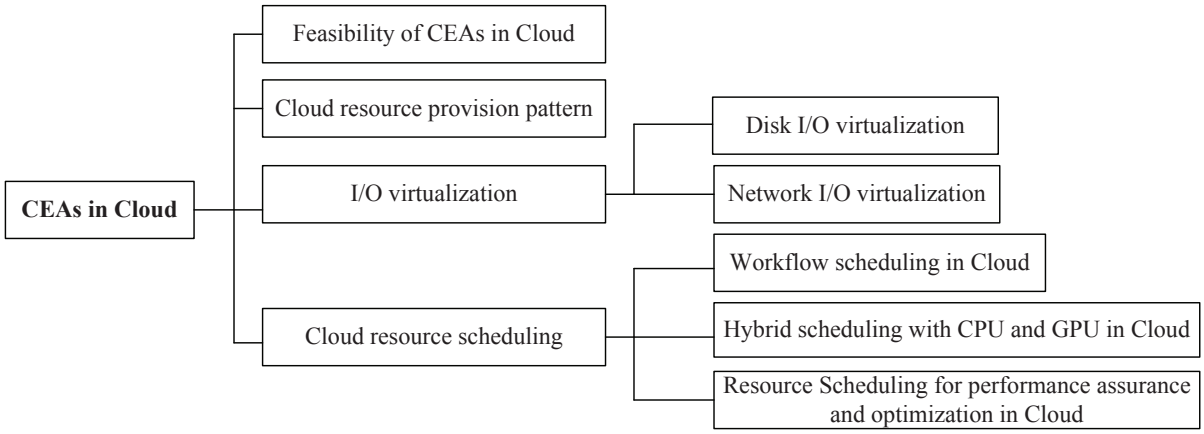


Fig. 2 Issues from the perspective of Cloud resource management when CEAs running in Cloud.

4 State of the Art

Since the CEA is the newcomer in Cloud computing system, in this section, we first survey the feasibility of running CEAs in Clouds. According to the characteristics of CEAs (described in Section 2), we then focus on three important aspects (i.e., resource provision pattern of Cloud, I/O virtualization technology, and Cloud resource scheduling) from the perspective of Cloud resource management.

Specifically, the organization of the state of the art in this section is shown in Fig. 2, and the reasons are as follows.

- In order to meet the diversified resource demands of CEAs in cloud, which is one of the CEAs' characteristics, we explore the research progress of the Cloud resource provision pattern.
- Components of CEAs are usually parallel-optimized programs. Therefore, I/O virtualization technology in Cloud will play an important role in the performance of CEAs when CEAs move to Cloud.
- Cloud resource scheduling is always an overarching research issue, especially for CEAs with complex internal executable process and many connected components.

4.1 Feasibility of CEAs Running in Cloud

CEAs usually require scientific computing and workflow. The recent works in [15–20] have showed feasibility of scientific computing applications and workflows in Cloud, and given some study on two aspects of "performance" and "cost". And the needs of the scientific Cloud are pointed out in [21], e.g., accessing parallel file systems, low-latency high bandwidth interconnect, legacy data sets, programming model for characteristics of scientific data and analysis methods, pre-installing and pre-tuning application software stacks, and customizing site-specific policies.

Besides, researchers explore the GPU acceleration for Computer Aided Engineering (so far, the closest example of CEAs) in GPU-enabled Amazon EC2 [22]. Their results show that, although good performance can be achieved, some development is still needed to achieve peak performance.

Summary: Cloud computing has emerged as the latest and most dominant utility computing solution. Existing researches show that it is feasible to run CEAs in cloud. On the other hand, they also indicate that the current cloud computing is not perfectly suitable to CEAs, because it is not specifically designed to perform CEAs.

4.2 Resource Provision Model of Cloud

CEAs typically run in dedicated clusters or HPC centers. In this section, we organize the state-of-the-art as follows. We first compare the resource provision models of traditional HPC centers and Cloud datacenters. And then we explore the requirements of CEAs on Cloud resources provision model.

The provision model of traditional HPC center resources is computing node pool. HPC center usually provides batch scheduling model to access computing resources. Its resource scheduling generally takes physical node as provision unit, because it is difficult to achieve isolation of different jobs on physical node. In order to serve different types of users, some HPC systems (such as IBM SP2 and HP Superdome) partition the compute nodes pool [23,24]. By contrast, Cloud datacenters based on virtualization technology organize resource to form virtual computing and storage pools. Taking Amazon EC2 as an example, customers can apply for cluster compute instances (CCI) to expedite their HPC workloads on elastic resources and save money by choosing low-cost pricing models [25].

Many CEAs have parallel computing components with intensive communication. If we run CEAs in current Amazon EC2 CCIs, their performance will suffer a lot from virtualization and network infrastructure [26]. In this case, a hybrid resource provision model composed of traditional HPC clusters and virtualized datacenters can better meet the needs of CEAs. Actually, different kinds of applications can obtain optimized performance by using different types of nodes in datacenters. For example, a study on constructing datacenter servers with general-purpose high-performance CPUs and low-power CPUs is carried out in [27] and [28], which reveal the possibility and necessity of heterogeneous resource provision in datacenters.

Summary: Recent studies show the probability and necessity of using heterogeneous resources to satisfy different kinds of applications in clouds. But up till now, it lacks systematic provision models of cloud resources for CEAs.

4.3 I/O Virtualization Technology in Cloud

I/O virtualization technology plays an important role in the performance of applications with intensive I/O requirements running in Cloud. However, the driver domain-based model for I/O virtualization exhibit poor performance. In this section, we investigate the related work on network and disk I/O virtualization in Cloud, respectively.

4.3.1 Network I/O virtualization

Over the last few years, a fair number of research efforts have been dedicated to enhance I/O virtualization technology so as to improve networking performance. Achievements can be divided into two classes: 1) software enhancements of driver domain model; and 2) advanced hardware enabled I/O virtualization.

For the software enhancements of driver domain model, the authors of [29] propose several optimizations to the memory sharing mechanism implemented in Xen so as to improve cache locality by moving grant copy operation from driver domain to guest VM. Besides, they propose to relax memory isolation property to reduce the number of grant operations performed. However, performance would come at the cost of isolation, one of the most attractive benefits of Xen architecture. Later on, the authors of [30] modify Xen to support multi-queue network interfaces to eliminate the software overheads of packet demultiplexing and copying and develop a grant reuse mechanism to reduce memory protection overheads. However, their work complicates live migration of VMs. A lightweight communication mechanism is proposed in [31] to mitigate the network I/O virtualization overhead of inter-domain communication and non-inter-domain communication. However, this work does not give the method of how to determine the relationship (i.e., inter-domain or non-inter-domain) between communicating VMs. In [32], the authors propose a new design for the memory sharing mechanism with Xen which completes the mechanism presented in [29]. The basic idea of the new mechanism is to enable the guest domains to unilaterally issue and revoke a grant. This allows the guest domains to protect their memory from incorrect direct memory access (DMA) operations. While such optimizations appear ideal for server-oriented workloads, the TCP/IP stack imposes a significant overhead when used for message passing.

Beyond the memory sharing mechanism, the authors of [33] propose to optimize the interrupt deliver route and shorten the network I/O path so as to increase the network I/O performance in virtualized environments. However, they point out that the performance impact is still substantial, especially on the receive side. In [34], the authors show how the cache topology greatly influences the performance. Then they propose to separate the interrupt handling from application processes and execute them on cores with cache affinity. Cache aware scheduling is proposed in [35] to reduce inter-domain communication cost. Furthermore, the authors improve packet processing in driver domain through a more efficient packet switching in a bridge. However, it suffers from the overhead of extra data copy. Recently, researchers at IBM Research laboratory present ELI (ExitLess Interrupts), a software-only approach for handling interrupts within guest VM directly and securely in order to improve the throughput and latency of unmodified guest operating systems [36]. The current ELI implementation configures the shadow IDT (Interrupt Descriptor Table) to force an exit when the guest is not supposed to handle an incoming physical interrupt, which results in the overhead of physical interrupts not assigned to the guest.

Since bridging operations inside driver domain introduce increasingly significant overhead when the number of VMs increases, several solutions of advanced hardware enabled I/O virtualization emerge to address this issue. Intel proposes using VM device queues (VMDq) [37]. At the chipset level, VMDq handles parallel queues of packets, routing them to the appropriate VMs and offloading the Virtual Machine Monitor (VMM). With concurrent direct network access (CDNA) technique [38, 39], the driver domain disappears from the networking architecture. The VMM directly assigns ownership of queues in the NIC to the VMs and deliv-

ers virtual interrupts upon arrival of packets to the appropriate destinations. Another hardware-based approach is SR/MR-IOV [40], a standard that allows a physical device to present itself to the system as multiple virtual devices, exporting to VMs part of the capabilities of smart network devices. Thus all the overhead related to the driver domain is eliminated. However, these techniques loose the traffic monitoring possibilities offered by the driver domain model.

4.3.2 Disk I/O virtualization

Disk I/O scheduler is an important part of modern operating systems. Its function is to improve disk utilization, gain better application performance, and performance isolation. Currently, the design of disk I/O scheduler is based on the assumption of the delay characteristics of underlying disk technology. In virtualized environment, VMM is used to share underlying storage resource among multiple competing VMs, and the delay characteristics of disk service observed in VM are very different from the assumption about delay characteristics in native environment. Therefore, it is necessary to redesign disk I/O scheduler of VMs. In this section, we survey two important issues of common concern on disk I/O virtualization as follows.

- disk bandwidth utilization;
- disk I/O resource allocation.

On the disk bandwidth utilization side, the work [41] presents Antfarm that can be used by a VMM to independently overcome part of the "semantic gap" separating it from the guest operating systems it supports, and improve disk bandwidth utilization in a virtualized environment. Antfarm relies on passive monitoring, which means that it is unable to guarantee interposition on events before they happen. A two-level scheduling framework, which decouples throughput and latency allocation to provide QoS guarantees to VMs while maintaining high disk utilization, is presented [42]. Solely based on requests' own characteristics, a light-weight disk scheduling framework [43] can make any work-conserving scheduler non-work-conserving, i.e., able to take future requests as dispatching candidates, to fully exploit locality and improve the utilization of disk bandwidth. In order to improve the disk performance of VMs, researchers from Peking University propose some optimizations: 1) reducing VM Exits by merging successive I/O instructions and decreasing the frequency of timer interrupt; and 2) removing some redundant operations from guest operating system, for example, the operations useless in virtual environment [44]. In [45], spatial locality is explored to improve disk efficiency in virtualized environments. The work in [46] presents a method for multiplexing multiple concurrent burst workloads on a shared storage server. In this work, VMM dynamically allocates I/O bandwidth among multiple VMs sharing a Storage Area Network (SAN) system. These methods are very helpful, however, they need to be extended to take the advantage of the availability of new technologies, such as solid state drive (SSD).

On disk I/O resource allocation side, current research mainly includes two aspects as follows.

- Disk I/O resource allocation based on fairness and performance isolation among VMs;
- Proportional allocation of disk I/O resource based on VM I/O performance requirements.

For the former aspect, the work in [47] presents a virtual I/O scheduler (VIOS) that provides absolute performance virtualization by fairly sharing I/O system resources among guest operating systems and their applications, and guarantees performance isolation in face of feature variations of I/O streams. On the other hand, it also demonstrates improved fairness at the cost of throughput. Researchers from Georgia Institute of Technology propose the notion of Differential Virtual Time (DVT), which can provide service differentiation with performance isolation for resource management mechanisms in guest operating system [48]. However, the improvement is still moderate compared to the available I/O capacity because, it does not explicitly focus on reducing the most important and common component of I/O processing workflow, i.e., IRQ processing latency.

For the latter aspect, researchers from VMware Inc. introduce PARDA [49], a software system that enforces proportional-share fairness among distributed hosts accessing a storage array, without assuming any support from the array itself. However, it does not take account of data locality. They also propose mClock [50], an algorithm for I/O resource allocation in a VMM, which supports proportional-share fairness subject to minimum reservations and maximum limits on the I/O allocations for VMs. However, the inherent tension between efficiency and isolation means that, in practice, cloud computing systems continue to provide poor isolation. The work in [51] presents FAIRIO, a cycle-based I/O scheduling algorithm. It enforces proportional sharing of I/O service through fair scheduling of disk time. During each cycle of the algorithm, I/O requests are scheduled according to workload weights and disk-time utilization history. Note, however, that a guaranteed disk-time share may not, in general, translate to a guaranteed throughput share. This is because two workloads with equal disk-time shares may receive

different bytes per second (B/s) or I/O operations per second (IOPS) due to the different costs of I/O requests and intra-disk scheduling decisions that may cause requests to be processed in an order deemed to maximize aggregate throughput. Researchers from Princeton University introduce Pisces [52], a system that achieves per-tenant weighted fair sharing of system resources across the entire shared service, even when partitions belonging to different tenants are co-located and when demand for different partitions is skewed or time-varying.

Summary: Cloud computing has gone deep in the hearts of people in the past few years. However, advances that make Cloud computing possible are not homogeneous: while adding more computational power was demonstrated feasible both in terms cost and scalability, the I/O abilities in terms of networking and storage are lagging behind. Given the data-intensive nature of CEA workloads, the performance of I/O virtualization is a significant factor in the overall performance of CEAs, thus becoming a critical focus area.

4.4 Resource Scheduling Methods in Cloud

Resource scheduling is the key focus of Cloud computing, and its policy and algorithm have a direct effect on the performance of applications. In this section, we survey related work on workflow scheduling in Cloud, because it is one of the effective ways to improve the efficiency of CEAs with complex internal processes. Since different users probably have different performance and resources requirements for CEAs, we summarize the related work on hybrid scheduling with CPU and GPU in Cloud and resource scheduling for the assurance and optimization of applications performance, respectively.

4.4.1 Workflow scheduling in Cloud

First, we present the limitation of workflow scheduling algorithms in traditional scenario. And then we summarize the research on workflow scheduling in Cloud.

In traditional scenario, most existing workflow scheduling algorithms only consider the computing environment in which the number and scale of compute resources is bounded. Compute resources in such an environment usually cannot be provisioned or released on demand, and these resources are not released to the environment until an execution of the workflow completes.

To address the above problem, a SHEFT workflow scheduling algorithm to schedule a workflow elastically in Cloud is proposed in [53]. Based on iterative ordinal optimization (IOO), the work in [54] proposes a workflow scheduling method to schedule scientific workflows in Clouds, and this method outperforms the Monte Carlo and Blind-Pick methods and yields higher performance against rapid workflow variations. A dynamic critical-path-based adaptive workflow scheduling algorithm for grids and Clouds is presented in [55], which determines efficient mapping of workflow tasks to grid and Cloud resources dynamically by calculating the critical path in the workflow task graph at every step. To improve the performance of running scientific workflows and cut down their cost in Clouds, a Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses is introduced in [56], which enables reliable and highly scalable execution of sequencing analyses workflows in a fully automated manner. And the work in [57] puts forward a two-stage data placement strategy and a task scheduling strategy for efficient workflow execution across geo-distributed datacenters by utilizing data dependencies.

During the data-intensive scientific workflow execution, a large volume of new intermediate data will be generated [58]. They can be even larger than the original data and contain some important intermediate results [59]. After the execution, some intermediate data need to be stored for future use so as to save regeneration cost when they are reused. Therefore, [59] presents an intermediate data storage strategy that can reduce the cost of scientific Cloud workflow system by automatically storing the most appropriate intermediate datasets into Cloud storage. In order to reduce the time cost of data movements between datacenters and deal with the data dependencies while keeping a relative load balancing of datacenters, the work in [60] proposes a data placement strategy.

4.4.2 Hybrid scheduling with CPU and GPU in Cloud

Because of powerful computing capabilities as well as relatively lower power consumption and cost, using accelerators for general-purpose computing is rapidly emerging in recent years. The coupling of accelerators and multi-core CPUs has given rise to a heterogeneous environment with a deep storage architecture (having the characteristics of non-uniform memory access) and a variety of computing devices. Here we focus on three aspects as follows.

- Hybrid scheduling with CPU and GPU;
- GPU virtualization;
- GPU in CAE/CAD.

In the research areas of hybrid scheduling with CPU and GPU, Mars [61] and Merge [62] are the first two projects to evaluate the joint use of GPU and CPU aiming the efficient execution of applications. Mars, a MapReduce framework

on the GPU is designed and implemented in [61], which provides a small set of APIs that are similar to those of CPU-based MapReduce. Mars was the first large scale GPU system, though its scalability is limited; it uses only one GPU and in-GPU-core tasks. Another shortcoming is that the library, not the user, schedules threads and blocks, making it hard to fully exploit certain GPU capabilities (e.g. inter-block communication). The Merge framework [62] replaces current ad hoc approaches to parallel programming on heterogeneous platforms with a rigorous, library-based methodology that can automatically distribute computation across heterogeneous cores to achieve increased energy and performance efficiency. With Merge framework, the computation-to-processor mappings are statically determined by the programmer. This manual approach not only puts burdens on the programmer but also is not adaptable, since the optimal mapping is likely to change with different applications, different input problem sizes, and different hardware/software configurations. The work in [63] presents a method for automated instantiation of heterogeneous FastFlow CPU/GPU parallel pattern applications on heterogeneous hybrid Cloud platforms. A systematic methodology is proposed in [64] to exploit approximated analytical performance/cost models, and an integrated programming framework that is suitable for targeting both local and remote resources to support the offloading of computations from structured parallel applications to heterogeneous Cloud resources. An experimental heterogeneous programming system called Qilin is implemented in [65], which proposes a fully automatic technique to map computations to processing elements on a machine equipped with CPU and GPU. In fact, Qilin requires and strongly relies on its own programming interface. This implies that the system cannot directly port the existing CUDA codes, but rather programmers should modify the source code to fit their interfaces. The work in [66] propose a compiler (generating middleware API code for the multi-core, as well as CUDA code to exploit the GPU) and runtime framework that can map a class of applications. While purely dynamic approaches neither require user intervention nor profiling, they do not take any kernel characteristics into account which often leads to poor performance [67].

On the GPU virtualization side, recent research focuses on how to make VM access to GPU accelerator [68–73]. GViM, a system designed for virtualizing and managing the resources of a general purpose system accelerated by graphics processors, is presented in [68]. GViM requires modification to the guest VMs running on the virtualized platform, a custom kernel module must be inserted to the guest operating system. The work in [69] describes vCUDA, which allows applications executing within VMs to leverage hardware acceleration and can be beneficial to the performance of some HPC applications in a transparent way. The GPU Virtualization Service (gVirtuS) presented in [70] tries to fill the gap between in-house hosted computing clusters, equipped with GPGPUs devices, and pay-for-use virtual clusters deployed via public or private computing Clouds. It implements various communicator components (TCP/IP, VMCI for VMware, VMSocket for KVM) to connect the front-end in guest OS and back-end in host OS. The design of communicator seems similar to the transfer channel in VMRPC [74], but RPC-based optimization that exploits the inherent semantics of the application, such as data presentation, would perform better than those merely optimizing data transfer channels. The authors of [75] propose a method of virtualizing high-end GPGPUs on ARM clusters for the next generation of high performance Cloud computing.

The application of GPU in CAE and CAD has been demonstrated by existing studies. A parallel explicit finite element (FE) based on GPU architecture for sheet forming is developed in [76] to address the tradeoff problem between the accuracy and efficiency for application introduced by the increase of complexity and scale of CAE model. A Cloud scheduler for finite element analysis service is presented in [77], which can dynamically select some of the required parameters, partition the load and schedule it in a resource-aware manner. Researchers from Argonne National Laboratory describe a GPU-based approach [78] for the dynamic simulation of complex CAE models where large collections of rigid bodies interact mutually through millions of frictional contacts and bilateral mechanical constraints.

The appearance of crack/gap artifacts between faces due to the approximations of the trimming curves prevents the usage of direct GPU rendering in CAD systems, and the work in [79] present a solution, combining a unique representation of the faces of the solid model with fragment shader algorithms. In [80], researchers apply GPU computing technology to carry out the machining error control which is a critical and time-consuming issue in 5-axis flank milling of complex geometries. And the numerical results of [81] show that the GPU architecture fits the PSO framework well by reducing computational timing, achieving high parallel efficiency and finding better optimal solutions by using a large number of particles. In [82], a GPU is used to effectively perform multi-body dynamic simulation with particle dynamics, and the results show that the greater the number of particles, the more computing time can be saved.

4.4.3 Resource Scheduling for performance assurance and optimization in Cloud

For the performance assurance of Cloud application, in [83], an autonomic resource manager is proposed. It can automate the management of virtual servers while taking into account both high-level QoS requirements of hosted applications and resource management costs. The work in [84] develops Q-Clouds, a QoS-aware control framework that tunes resource allocations to mitigate performance interference effects. A service management framework implementation is defined in [85], which supports on demand Cloud provisioning and present a monitoring framework that meets the demands of Cloud-based applications. The service provisioning problem is modeled as a Generalized Nash game in [86], and an algorithm is proposed to manage and allocate the IaaS resources to the competing Software-as-a-Services (SaaSs) which need to comply with QoS requirements. Similarly, a multi-objective optimization genetic algorithm [87] is presented to decide how to allocate virtualized resources on the Cloud to Web applications. In order to implement the QoS-aware execution of cloud applications, a resource elasticity framework is presented in [88]. A SLA-driven resource scheduling scheme is designed in [89] that selects a proper datacenter among globally distributed centers operated by a provider. And a SLA-aware autonomic cloud solution is introduced in [90], which enables the management of large scale infrastructure while supporting multiple dynamic requirement from users.

A dynamic service provision (DSP) model [91], for many task computing (MTC) or high throughput computing (HTC) service providers is proposed by researchers at Chinese Academy of Sciences. Based on DSP model, the Dawningcloud is implemented, which provides automatic management for MTC and HTC workloads. In order to improve the performance of parallel applications with synchronization requirements in Cloud, a cache contention-aware scheduling approach is proposed in [92]. And a communication-driven scheduling approach for virtual clusters in cloud is presented in [93, 94].

Summary: Generally, Cloud resource scheduling plays an important role in the performance of Cloud applications. To assure and improve the performance of Cloud applications, researchers have proposed many different approaches for different types of applications in recent years. The emergence of new type of Cloud application, e.g., CEA discussed in this paper, brings new challenges of resources scheduling. It can motivate researchers to study new scheduling solutions, which should consider heterogeneity of the Cloud resources and the features of CEA workloads.

5 Challenges and Research Topics

CEAs usually involve a number of components with strong dependency [6, 95–98], and have the problems of iterative tuning. They not only have large amount of parallel computing and intermediate files, but also have high disk I/O load [19,58,99,100] and frequent communication [18]. Based on the related literature in Section 4, we find that three challenges have not been sufficiently solved when CEAs running in Cloud from the perspective of Cloud resource management. The overview of challenges and research topics is shown in Fig. 3.

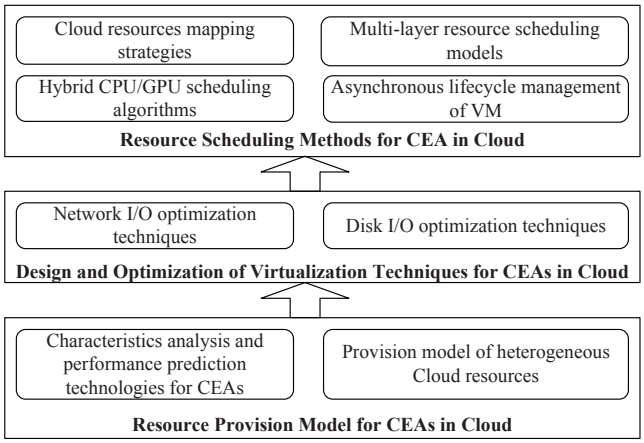


Fig. 3 Research topics of cloud resource management for CEAs

5.1 How to Build a New Cloud Resource Provision Model Based on the Resource Demands of CEA?

To obtain better performance for CEA in Cloud, one of the issues needed to be solved is that how Cloud computing provides appropriate resource provision model according to the resource demands characteristics of CEA. The resource provision model in traditional high performance computing center is based on physical node pool and the resource scheduling algorithms mainly consider when and how the requested computing nodes should be assigned to the jobs in job queue. However, the existing Cloud computing centers usually use virtualization technology to partition the physical nodes into many independent VMs, and provide virtual storage devices such as Amazon Simple Storage Service (S3) and Elastic Block Storage (EBS).

The requirement of new resource provision model arises when CEAs run in Cloud. The reasons are as follows: First,

CEAs often require a large amount of computation and low-latency communication, but the VM technology increases communication delay obviously. That is to say, VM-based organization form of traditional Cloud computing resources may not provide full support for CEAs. Second, different components of CEAs may vary a lot in the demand for resources. For example, some components may be compute-intensive, and some others may be I/O-intensive. So CEAs are likely to put forward the demand for heterogeneous nodes in Cloud. However, the provision model of resources in the existing Clouds mainly focuses on approximate homogeneous nodes.

In order to improve the performance of CEAs in Cloud, characteristics analysis and performance prediction technologies for CEAs and new resource provision models in Cloud are extremely needed.

5.1.1 Characteristics analysis and performance prediction technologies for CEAs running in Cloud

First, we need accurate metrics reflecting the characteristics of CEAs, and methods extracting the abstract descriptions of the affinities and dependencies among CEAs components. Second, we need to identify the resource objects which determine the performance of CEAs, then study the techniques of relationship mining between resource objects and application performance. Based on these work, performance prediction technology for CEAs in Cloud can be studied so as to lay the foundation for the research on Cloud resource provision and management methods towards CEAs.

5.1.2 Provision model of heterogeneous Cloud resources

Different components of CEA may be inclined to require different types of Cloud resources. In this situation, it is significant to explore the provision model of heterogeneous resources in Cloud. Based on the effective techniques of resource organization and allocation in traditional high performance computing environment and virtualization technology, researchers need to explore the appropriate expression for the heterogeneity of Cloud resource, and analyze the impact of different resources provision models on CEA performance, then study the resource provision models suitable for different types of CEAs in Cloud.

5.2 How to Design and Optimize Virtualization Techniques for CEA in Cloud Environment?

Virtualization benefits for system management and resource utilization. However, it introduces additional performance overhead. It cannot avoid the performance decrease of CEAs introduced by virtualization in current Cloud system, especially when CEAs contain a large number of disk or network I/O-intensive components, or a combination of both.

Taking disk I/O as an example, since VMM does not understand disk access characteristics of applications running in VM, VMM not only lacks of global view of disk access characteristics of processes, but also ignores the I/O access characteristics of VMs. Therefore, it results in inefficient low-level I/O scheduling and degrades the performance of CEAs.

There exists a contradiction between the high-performance requirements of CEAs and performance loss introduced by virtualization technology. The point is that we need to maintain the strengths of virtualization technology, while overcoming its negative impacts. Recently, some optimization methods for virtualization are put forward, but at the expense of sacrificing some virtualization merits (e.g., live migration, isolation, and security). Therefore, it is necessary to re-design and optimize the virtualization techniques in Cloud to satisfy the performance requirements of CEAs.

5.2.1 Network I/O optimization techniques

Parallel computing technology has been widely used in engineering applications to speed up the design process. Therefore, it is important to explore the application-aware allocation strategies and control techniques of network I/O latency and bandwidth. Besides, researchers also need to address the semantic isolation problem and reduce the overhead of context switching resulted from interrupting the critical path of VM network I/O, so as to optimize network I/O performance of Cloud computing system to meet the needs of CEA for low-latency network communication.

5.2.2 Disk I/O optimization techniques

Typically, the size of intermediate files generated in CEA workflow is very large. Targeting at this characteristic of intermediate files, researchers need to study the prediction mechanism which can reflect changes in the access pattern of VM disk, thereby contributing to the establishment of disk I/O scheduling algorithms for CEAs. Besides, researchers also need to explore dynamic control techniques of disk I/O bandwidth and delay in Cloud to meet the different demands

of CEA on disk access and storage performance.

5.3 How to Schedule Cloud Resources to Enhance CEA Performance and the Service Capacity of Cloud?

Traditional CEAs usually run in specialized cluster systems or HPC centers, where the running environment of CEAs is often dedicated or preconfigured, so it has few problems of dynamic resource scheduling. However, in CEA-oriented Cloud environment, CEAs users have multiple service models to choose: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service), and combinations of above three. Therefore, researchers have to face more complex problems of resource scheduling. For example, how to map the VMs to appropriate physical nodes? How to do life-cycle management for VMs? Briefly, all of these issues drive us to take an overall consideration on new resource scheduling models and methods.

5.3.1 Cloud resources mapping strategies

For CEA workflow running in Cloud, it needs models to capture the relationship between the CEA performance and the cost of resource, which can help achieve optimal matching between CEA and the available resources supported by virtualization technology. Besides, it also needs to explore remapping strategies of resources to adapt to dynamic changes in the resource demands of CEA.

5.3.2 Multi-layer resource scheduling models

In Cloud, resource scheduling usually involves many kinds of objects from multiple layers including VM, physical machine, multi-core processors, etc. In order to guarantee the QoS of CEA running in Cloud, it is important to study multi-layer scheduling models under the conditions of optimal resource mapping discussed above, which help make the best use of Cloud resources and improve the execution efficiency of CEAs.

5.3.3 Hybrid CPU/GPU scheduling algorithms

As mentioned, the resource requirements of CEA at different stages during run time are usually different. For example, some stages only require CPUs, and some others require CPUs and GPUs. Therefore, it is necessary to explore the strategies of resources allocation, resources reservation, task preemption, and task migration to schedule CPU and GPU tasks effectively. That is to say, researchers should study hybrid CPU/GPU scheduling algorithms to improve the performance of CEA workflow.

5.3.4 Asynchronous life-cycle management of VM

CEA workflow comprised of multiple stages typically requires more than one VM. Therefore, it is inevitable to encounter the problems of asynchronous life-cycle management of VMs. To address this issue, researchers need to explore how to identify and define the life-cycle of VMs, and to study how to implement fast deployment, cloning, launching, and destruction of numerous VMs. Besides, researchers also need to handle the persistent and temporary data in the life-cycle of VMs.

6 Conclusions

Cloud computing has recently emerged as a compelling paradigm for managing and delivering services over the Internet, which provides new chances for CEAs by low-cost massive computing and storage resource and customizable execution environment. However, it is proved that current Cloud resource management technologies are not matured or suitable for CEAs. After a comprehensive study, we give a survey on the state of art and summarize the research challenges on resource provision, virtualization, and resource scheduling which have significant impact on CEA's performance. At last, a series of research topics are proposed following these challenges. We believe there is still enough room for improving CEAs' performance running in Cloud. And there are tremendous opportunities for researchers to make ground-breaking contributions in this field.

Acknowledgements We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by National Science Foundation of China under grant No.61232008 and No.61472151, National 863 Hi-Tech Research and Development Program under grant No.2015AA01A203 and No.2014AA01A302, the Fundamental Research Funds for the Central Universities under grant No.2015TS067.

References

1. Engineering computing in cloud, `http://www-03.ibm.com/systems/technicalcomputing/solutions/cloud/engineering_solutions.html`
2. Crago S, Dunn K, Eads P, Hochstein L, Kang D I, Kang M, Modium D, Singh K, Suh J, Walters J P. Heterogeneous cloud computing. In: Proceedings of 2011 IEEE International Conference on Cluster Computing (CLUSTER). 2011, 378–385

3. Arian E. On the coupling of aerodynamic and structural design. Journal of Computational Physics, 1997, 135(1): 83–96
4. Wang C, Xu L. Parameter mapping and data transformation for engineering application integration. Information Systems Frontiers, 2008, 10(5): 589–600
5. Ong M, Thompson H. Challenges for wireless sensing in complex engineering applications. In: Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society (IECON). 2011, 2106–2111
6. Bichon B, Eldred M, Swiler L, Mahadevan S, McFarland J. Multimodal reliability assessment for complex engineering applications using efficient global optimization. In: Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA-2007-1946. 2007, 3029–3040
7. Chacón Rebollo T, Gómez Mármol M, Restelli M. Numerical analysis of penalty stabilized finite element discretizations of evolution navier–stokes equations. Journal of Scientific Computing, 2015, 63(3): 885–912
8. Campana E F, Liuzzi G, Lucidi S, Peri D, Piccialli V, Pinto A. New global optimization methods for ship design problems. Optimization and Engineering, 2009, 10(4): 533–555
9. Tribon, https://en.wikipedia.org/wiki/Tribon
10. ANSYS Fluent, http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/Fluid+Dynamics+Products/ANSYS+Fluent
11. ANSYS CFD-Post, http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics/Results+Analysis+Products/ANSYS+CFD-Post
12. Jin C, Wang Y, Zhang W, Lin Y. Study on semi-finished ship structural components assembly sequence optimization. In: Proceedings of 2010 Sixth International Conference on Natural Computation (ICNC). 2010, 2706–2709
13. Kwon S, Kim B C, Mun D, Han S. Simplification of feature-based 3D CAD assembly data of ship and offshore equipment using quantitative evaluation metrics. Computer-Aided Design, 2015, 59: 140–154
14. Full-service EDA Cloud, http://sicadinc.com/
15. Palankar M R, Iamnitchi A, Ripeanu M, Garfinkel S. Amazon S3 for science grids: a viable solution? In: Proceedings of the 2008 International Workshop on Data-aware Distributed Computing. 2008, 55–64
16. Hazelhurst S. Scientific computing using virtual high-performance computing: a case study using the amazon elastic computing cloud. In: Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology. 2008, 94–103
17. Vöckler J S, Juve G, Deelman E, Rynge M, Berriman B. Experiences using cloud computing for a scientific workflow application. In: Proceedings of the 2nd International Workshop on Scientific Cloud Computing. 2011, 15–24
18. Juve G, Deelman E, Vahi K, Mehta G, Berriman B, Berman B P, Maechling P. Scientific workflow applications on amazon EC2. In: Proceedings of 2009 5th IEEE International Conference on E-Science Workshops. 2009, 59–66
19. Rehr J J, Vila F D, Gardner J P, Svec L, Prange M. Scientific computing in the cloud. Computing in Science & Engineering, 2010, 12(3): 34–43
20. Lin G, Han B, Yin J, Gorton I. Exploring cloud computing for large-scale scientific applications. In: Proceedings of the IEEE 9th World Congress on Services (SERVICES). 2013, 37–43
21. Ramakrishnan L, Zbiegel P T, Campbell S, Bradshaw R, Canon R S, Coghlan S, Sakrejda I, Desai N, Declerck T, Liu A. Magellan: experiences from a science cloud. In: Proceedings of the 2nd International Workshop on Scientific Cloud Computing. 2011, 49–58
22. Georgescu S, Chow P. GPU accelerated CAE using open solvers and the cloud. ACM SIGARCH Computer Architecture News, 2011, 39(4): 14–19
23. OpenPBS, http://www.mcs.anl.gov/research/projects/openpbs/
24. Platform Inc. LSF, http://www.platform.com/workload-management/high-performance-computing
25. Amazon EC2, http://aws.amazon.com/ec2/
26. Zhai Y, Liu M, Zhai J, Ma X, Chen W. Cloud versus in-house cluster: evaluating amazon cluster compute instances for running MPI applications. In: Proceedings of The International Conference for High Performance Computing, Networking, Storage and Analysis (SC). 2011, 11
27. Janapa Reddi V, Lee B C, Chilimbi T, Vaid K. Web search using mobile cores: quantifying and mitigating the price of efficiency. 2010, 38(3): 314–325
28. Lim K, Ranganathan P, Chang J, Patel C, Mudge T, Reinhardt S. Understanding and designing new server architectures for emerging warehouse-computing environments. In: Proceedings of the 35th International Symposium on Computer Architecture(ISCA). 2008, 315–326
29. Santos J R, Turner Y, Janakiraman G J, Pratt I. Bridging the gap between software and hardware techniques for I/O virtualization. In: Proceedings of the USENIX Annual Technical Conference (ATC). 2008, 29–42
30. Ram K K, Santos J R, Turner Y, Cox A L, Rixner S. Achieving 10 Gb/s using safe and transparent network interface virtualization. In: Proceedings of the ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE). 2009, 61–70
31. Chen H, Wu S, Shi X, Jin H, Fu Y. LCM: A lightweight communication mechanism in HPC cloud. In: Proceedings of the 6th International Conference on Pervasive Computing and Applications. 2011, 443–451
32. Ram K K, Santos J R, Turner Y. Redesigning Xen's memory sharing mechanism for safe and efficient I/O virtualization. In: Proceedings of the 2nd conference on I/O virtualization. 2010, 1–1
33. Jian Z, Xiaoyong L, Haibing G. The optimization of Xen network virtualization. In: Proceedings of the International Conference on Computer Science and Software Engineering. 2008, 431–436

34. Guo D, Liao G, Bhuyan L N. Performance characterization and cache-aware core scheduling in a virtualized multi-core server under 10GbE. In: Proceedings of the IEEE International Symposium on Workload Characterization. 2009, 168–177
35. Liao G, Guo D, Bhuyan L, King S R. Software techniques to improve virtualized I/O performance on multi-core systems. In: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems. 2008, 161–170
36. Gordon A, Amit N, Har'El N, Ben-Yehuda M, Landau A, Schuster A, Tsafrir D. ELI: bare-metal performance for I/O virtualization. In: Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). 2012, 411–422
37. Abramson D. Intel virtualization technology for directed I/O. Intel Technology Journal, 2006, 10(3): 179–192
38. Rixner S. Network virtualization: Breaking the performance barrier. Queue, 2008, 6(1): 37:36–37:ff
39. Willmann P, Shafer J, Carr D, Menon A, Rixner S, Cox A L, Zwaenepoel W. Concurrent direct network access for virtual machine monitors. In: Proccedings of the IEEE 13th International Symposium on High Performance Computer Architecture (HPCA). 2007, 306–317
40. Liu J. Evaluating standard-based self-virtualizing devices: A performance study on 10 GbE NICs with SR-IOV support. In: Proceedings of the IEEE International Symposium onParallel & Distributed Processing (IPDPS). 2010, 1–12
41. Jones S T, Arpaci-Dusseau A C, Arpaci-Dusseau R H. Antfarm: Tracking processes in a virtual machine environment. In: Proceedings of the USENIX Annual Technical Conference (ATC). 2006, 1–14
42. Jin H, Ling X, Ibrahim S, Cao W, Wu S, Antoniu G. Flubber: Two-level disk scheduling in virtualized environment. Future Generation Computer Systems, 2013, 29(8): 2222–2238
43. Xu Y, Jiang S. A scheduling framework that makes any disk schedulers non-work-conserving solely based on request characteristics. In: USENIX Conference on File and Storage Technologies. 2011, 119–132
44. Zhang B B, Wang X L, Yang L, Lai R F, Wang Z L, Luo Y W, Li X M. Modifying guest OS to optimize I/O virtualization in KVM. Jisuanji Xuebao(Chinese Journal of Computers), 2010, 33(12): 2312–2319
45. Ling X, Ibrahim S, Jin H, Wu S, Tao S. Exploiting spatial locality to improve disk efficiency in virtualized environments. In: Proceedings of the IEEE 21st International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). 2013
46. Wang H, Varman P. A flexible approach to efficient resource sharing in virtualized environments. In: Proceedings of the 8th ACM International Conference on Computing Frontiers. 2011, 39
47. Seelam S R, Teller P J. Virtual I/O scheduler: a scheduler of schedulers for performance virtualization. In: Proceedings of the 3rd International Conference on Virtual Execution Environments. 2007, 105–115
48. Kesavan M, Gavrilovska A, Schwan K. Differential virtual time (DVT): rethinking I/O service differentiation for virtual machines. In: Proceedings of the 1st ACM symposium on Cloud computing. 2010, 27–38
49. Gulati A, Ahmad I, Waldspurger C A, others . PARDA: proportional allocation of resources for distributed storage access. In: Proccedings of the 7th Conference on File and Storage Technologies (FAST). 2009, 85–98
50. Gulati A, Merchant A, Varman P J. mClock: handling throughput variability for hypervisor I/O scheduling. In: Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI). 2010, 1–7
51. Arunagiri S, Kwok Y, Teller P J, Portillo R A, Seelam S R. FAIRIO: A throughput-oriented algorithm for differentiated I/O performance. International Journal of Parallel Programming, 2014, 42(1): 165–197
52. Shue D, Freedman M J, Shaikh A. Performance isolation and fairness for multi-tenant cloud storage. In: Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI). 2012, 349–362
53. Lin C, Lu S. Scheduling scientific workflows elastically for cloud computing. In: Proceedings of the IEEE International Conference on Cloud Computing (CLOUD). 2011, 746–747
54. Zhang F, Cao J, Hwang K, Wu C. Ordinal optimized scheduling of scientific workflows in elastic compute clouds. In: Proceedings of 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom). 2011, 9–17
55. Rahman M, Hassan R, Ranjan R, Buyya R. Adaptive workflow scheduling for dynamic grid and cloud computing environment. Concurrency and Computation: Practice and Experience, 2013, 25(13): 1816–1842
56. Liu B, Li J, Liu C. Cloud-based bioinformatics workflow platform for large-scale next-generation sequencing analyses. Journal of Biomedical Informatics, 2014, 49: 119–133
57. Shao-Wei L, Ling-Mei K, Jun k R, Jun-Qiang S, Ke-Feng D, Hong-Ze L. A two-step data placement and task scheduling strategy for optimizing scientific workflow performance on cloud computing platform. Chinese Journal of Computers, 2011, 34(11): 2121–2130
58. Deelman E, Chervenak A. Data management challenges of data-intensive scientific workflows. In: Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid (CCGRID). 2008, 687–692
59. Yuan D, Yang Y, Liu X, Chen J. A cost-effective strategy for intermediate data storage in scientific cloud workflow systems. In: Proceedings of 2010 IEEE International Symposium on Parallel and Distributed Processing (IPDPS). 2010, 1–12
60. Pai Z, Li-Zhen C, Hai-Yang W, Meng X. A data placement strategy for data-intensive applications in cloud. Chinese Journal of Computers, 2010, 33(8): 1472–1480
61. He B, Fang W, Luo Q, Govindaraju N K, Wang T. Mars: a mapreduce framework on graphics processors. In: Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques (PACT). 2008, 260–269
62. Linderman M D, Collins J D, Wang H, Meng T H. Merge: a pro-

gramming model for heterogeneous multi-core systems. In: Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). 2008, 287–296

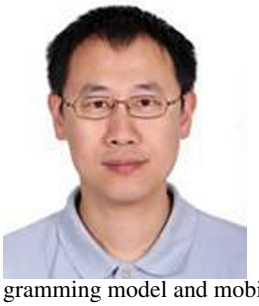
63. Boob S, Gonzalez-Velez H, Popescu A M. Automated instantiation of heterogeneous fast flow CPU/GPU parallel pattern applications in clouds. In: Proceedings of the 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). 2014, 162–169
64. Campa S, Danelutto M, Goli M, González-Vélez H, Popescu A M, Torquati M. Parallel patterns for heterogeneous CPU/GPU architectures: Structured parallelism from cluster to cloud. Future Generation Computer Systems, 2014, 37: 354–366
65. Luk C K, Hong S, Kim H. Qilin: exploiting parallelism on heterogeneous multiprocessors with adaptive mapping. In: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). 2009, 45–55
66. Ravi V T, Ma W, Chiu D, Agrawal G. Compiler and runtime support for enabling generalized reduction computations on heterogeneous parallel configurations. In: Proceedings of the 24th ACM International Conference on Supercomputing (ICS). 2010, 137–146
67. Grewe D, O'Boyle M F. A static task partitioning approach for heterogeneous systems using OpenCL. In: Compiler Construction. 2011, 286–305
68. Gupta V, Gavrilovska A, Schwan K, Kharche H, Tolia N, Talwar V, Ranganathan P. GViM: GPU-accelerated virtual machines. In: Proceedings of the 3rd ACM Workshop on System-level Virtualization for High Performance Computing. 2009, 17–24
69. Shi L, Chen H, Sun J, Li K. vCUDA: GPU-accelerated high-performance computing in virtual machines. IEEE Transactions on Computers, 2012, 61(6): 804–816
70. Giunta G, Montella R, Agrillo G, Coviello G. A GPGPU transparent virtualization component for high performance computing clouds. In: Proceedings of Euro-Par 2010-Parallel Processing, 379–391. Springer, 2010
71. Jo H, Jeong J, Lee M, Choi D H. Exploiting GPUs in virtual machine for BioCloud. BioMed Research International, 2013, 2013
72. Shih C S, Wei J W, Hung S H, Chen J, Chang N. Fairness scheduler for virtual machines on heterogonous multi-core platforms. ACM SIGAPP Applied Computing Review, 2013, 13(1): 28–40
73. Hu L, Che X, Xie Z. GPGPU cloud: A paradigm for general purpose computing. Tsinghua Science and Technology, 2013, 18(1): 22–23
74. Chen H, Shi L, Sun J. VMRPC: A high efficiency and light weight RPC system for virtual machines. In: The 18th IEEE International Workshop on Quality of Service (IWQoS). 2010
75. Montella R, Giunta G, Laccetti G. Virtualizing high-end GPGPUs on ARM clusters for the next generation of high performance cloud computing. Cluster Computing, 2014, 17(1): 139–152
76. Cai Y, Li G, Wang H, Zheng G, Lin S. Development of parallel explicit finite element sheet forming simulation system based on GPU architecture. Advances in Engineering Software, 2012, 45(1): 370–379
77. Ari I, Muhtaroglu N. Design and implementation of a cloud computing service for finite element analysis. Advances in Engineering Software, 2013, 60: 122–135
78. Negrut D, Tasora A, Anitescu M, Mazhar H, Heyn T, Pazouki A. Solving large multi-body dynamics problems on the GPU. GPU Gems, 2011, 4: 269–280
79. Hanniel I, Haller K. Direct rendering of solid CAD models on the GPU. In: Proceedings of the 12th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics). 2011, 25–32
80. Hsieh H T, Chu C H. Particle swarm optimisation (PSO)-based tool path planning for 5-axis flank milling accelerated by graphics processing unit (GPU). International Journal of Computer Integrated Manufacturing, 2011, 24(7): 676–687
81. Hung Y, Wang W. Accelerating parallel particle swarm optimization via GPU. Optimization Methods and Software, 2012, 27(1): 33–51
82. Jung H Y, Jun C W, Sohn J H. GPU-based collision analysis between a multi-body system and numerous particles. Journal of Mechanical Science and Technology, 2013, 27(4): 973–980
83. Nguyen Van H, Dang Tran F, Menaud J M. Autonomic virtual resource management for service hosting platforms. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. 2009, 1–8
84. Mehta H K, Kanungo P, Chandwani M. Performance enhancement of scheduling algorithms in clusters and grids using improved dynamic load balancing techniques. In: Proceedings of the 20th International Conference Companion on World Wide Web (WWW). 2011, 385–390
85. Chapman C, Emmerich W, Márquez F G, Clayman S, Galis A. Software architecture definition for on-demand cloud provisioning. Cluster Computing, 2012, 15(2): 79–100
86. Ardagna D, Panicucci B, Passacantando M. A game theoretic formulation of the service provisioning problem in cloud systems. In: Proceedings of the 20th International Conference Companion on World Wide Web (WWW). 2011, 177–186
87. Qiang L, Qin-Fen H, Li-Min X, Zhou-Jun L. Adaptive management and multi-objective optimization for virtual machine placement in cloud computing. Chinese Journal of Computers, 2011, 34(12): 2253–2264
88. Kaur P D, Chana I. A resource elasticity framework for QoS-aware execution of cloud applications. Future Generation Computer Systems, 2014, 37: 14–25
89. Son S, Jun S C. Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing. In: Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). 2013, 168–171
90. García García A, Blanquer Espert I, Hernández García V. SLA-driven dynamic cloud resource management. Future Generation Computer Systems, 2014, 31: 1–11
91. Wang L, Zhan J, Shi W, Liang Y, Yuan L. In cloud, do mtc or htc service providers benefit from the economies of scale? In: Proceedings of the 2nd Workshop on Many-Task Computing on Grids and

Supercomputers. 2009, 7
92. Jin H, Qin H, Wu S, Guo X. CCAP: A cache contention-aware virtual machine placement approach for HPC cloud. International Journal of Parallel Programming, 2015, 43(3): 403–420
93. Chen H, Wu S, Di S, Zhou B, Xie Z, Jin H, Shi X. Communication-driven scheduling for virtual clusters in cloud. In: Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing (HPDC), HPDC '14. 2014, 125–128
94. Wu S, Chen H, Di S, Zhou B, Xie Z, Jin H, Shi X. Synchronization-aware scheduling for virtual clusters in cloud. IEEE Transactions on Parallel and Distributed Systems, 2015, 99(PrePrints): 1
95. Eldred M S. Optimization strategies for complex engineering applications. Technical report, Sandia National Labs., Albuquerque, NM (United States), 1998
96. Keahey K. Cloud computing for science. In: Proceedings of the 21st International Conference on Scientific and Statistical Database Management. 2009, 478
97. Deelman E, Singh G, Livny M, Berriman B, Good J. The cost of doing science on the cloud: the montage example. In: Proceedings of the 2008 ACM/IEEE conference on Supercomputing (ICS). 2008, 50
98. Wang L, Tao J, Kunze M, Castellanos A C, Kramer D, Karl W. Scientific cloud computing: Early definition and experience. In: Proceedings of the IEEE International Conference onHigh Performance Computing and Communications. 2008, 825–830
99. Ludäscher B, Altintas I, Berkley C, Higgins D, Jaeger E, Jones M, Lee E A, Tao J, Zhao Y. Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience, 2006, 18(10): 1039–1065
100. Oinn T, Addis M, Ferris J, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock M R, Wipat A, others . Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics, 2004, 20(17): 3045–3054

Haibao CHEN is currently working toward the PhD degree in Service Computing Technology and System Lab (SCTS) and Cluster and Grid Lab (CGCL) at Huazhong University of Science and Technology (HUST) in China. He is also with the Big Data and Cloud Lab, School of Computer and information engineering, Chuzhou University. His research interests include parallel and distributed computing, virtualization, and resource scheduling on cloud computing.

Song WU is a professor of computer science and engineering at Huazhong University of Science and Technology (HUST) in China. He received his Ph.D. from HUST in 2003. His current research interests include cloud computing, system virtualization, datacenter management, storage system, in-memory computing and so on.

Hai JIN received the PhD degree in computer engineering from HUST in 1994. He is the chief scientists of National 973 Basic Research Program Project of Virtualization Technology of Computing System, and Cloud Security. His research interests include computer architecture, virtualization technology, cluster computing and cloud computing, peer-to-peer computing, network storage, and network security.



Wenguang CHEN received the B.S. and Ph.D. degrees in computer science from Tsinghua University in 1995 and 2000 respectively. He is a professor in Department of Computer Science and Technology, Tsinghua University. His research interest is in parallel and distributed computing, programming model and mobile cloud computing.

Jidong ZHAI is an assistant professor at the Department of Computer Science and Technology, Tsinghua University. His research focuses on high performance computing, especially performance analysis and optimization for large-scale parallel applications and performance evaluation for computer systems. He received his PhD degree from Tsinghua University in 2010.

Yingwei LUO received his B.S. degree from Zhejiang University in 1993, and his M.S. and Ph.D. degrees from Peking University in 1996 and 1999 respectively. He is a professor in Peking University. His research interests include virtualization technologies, distributed computing and geographic information system, etc.

Xiaolin WANG received his B.S. and Ph.D. degrees from Peking University in 1996 and 2001 respectively. He is an associate professor in Peking University. His research interests include virtualization technologies, distributed computing and geographic information system, etc.