软件定义的云际存储

沃天宇 杨任宇 王媛冬 等
北京航空航天大学

关键词：分布式系统 云际计算 云际存储

引言

云计算在过去几年已成为促进互联网新兴行业创新发展的重要支撑。随着“互联网 +”概念和战略的深化，互联网技术和概念与各个传统行业出现了更为深入的融合，通过互联网平台，互联网技术已经创造出更多更大的生态链 [1]。然而，单一云平台难以满足云应用要求的全时全域的优质服务需求和灵活多样的个性化定制需求。

以云服务实体之间开放协作为基础，通过多方云资源深度融合进行多云之间互联的**云际计算**将成为新一代云计算模式。由于数据的高效存储在大数据处理、大规模数据服务支撑中会起到至关重要的作用，云际存储将作为云际计算的重要组成部分。为实现广域网环境下多云之间协作的云际服务，软件定义的方法可以将一体化的应用设计细粒度拆分，通过硬件虚拟化为上层软件层提供应用编程接口 (API)，软件层则可以实现灵活高效的管理与控制 [2]。软件定义的存储能够创建跨地理分布式的存储资源池，将云中所有物理存储设备转化为统一、虚拟、共享的存储资源池中的基本单元，同时，将存储的控制与管理从物理设备中抽象与分离，形成硬件与上层应用之间的软件层。利用云际计算中云际互联的资源能够提供更为安全可靠的系统服务，提高全局的能效优化、降低运营成本；在不同的云提供商上部署应用，可以减少单一数据中心中相关性故障 (correlated failure) 的发生概率；通过软件定义的策略制定、配置的自适应适配，能够最大限度地减少存储服务的请求延迟，避免短时请求波动与蜂拥及其他网络不确定性对服务质量造成的负面影响。

应用场景与挑战

为了更好地理解软件定义的云际存储的功能技术细节，我们首先讨论一些典型的云际存储应用场景，进而总结云际存储的需求与挑战。

典型案例分析

用户透明的多云盘协作 无论何时、身在何处、使用哪种终端设备（个人电脑、平板电脑、智能手机）、使用哪种云存储服务 (Dropbox, Google Drive, Microsoft OneDrive)，都能为用户提供数据获取和共享功能，这就是多云盘协作。云存储服务支持多种数据的存储，包括文件、图片、音乐、视频等，均能在用户的各个设备之间在线同步，并且能够与他人分享。然而，云存储服务聚合场景中面临着访问接口异构、存储特征异构、地理分布、数据强一致性等方面的挑战，对容量聚合、突发需求变化、访问性能优化、服务可用性等方面也有着迫切需求。

车联网系统中多源数据融合与溯源 如今，通过车联网手机客户端约车已成常事。智能交通出行规划、位置数据、传感器数据、车辆数据以及诸多海量多源数据都需要高效存储与共享。对这些数据的挖掘和分析，对于交通流量管理、出租车调度等起着至关重要的作用。尤其对不同特定云的不同数据源加以融合能够更好地满足以上需求。例如，

交通云（包含准确的路网信息）和气象云（包含大量的气象预报信息）能够丰富出行规划的设计，提供更精确智能的行程预测。

医疗健康系统 电子健康档案(Electronic Health Records, EHRs)[3]是由多个不同种类的医疗机构经过长期积累逐步建立起来的。这些医疗机构要求实时掌握病人的健康信息，在必要时对病人以往的健康信息（如病历、检测结果等）进行综合分析，以做出最好的病情判断。因此，云际存储在该场景中面临着数据分布广、数据共享的高效对等、隐私保护等挑战，对数据共享的认证、激励与授权、数据的同步和一致性等方面的需求尤其紧迫。

需求与挑战

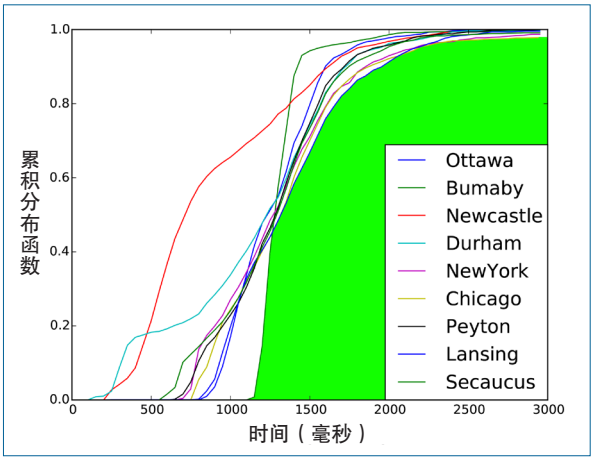

图1 同一服务来自不同位置的调用的响应延迟

弹性的存储容量与存储性能 通过利用分散在不同云服务提供者的资源，存储服务需要在不同的条件和需求下，实现全局的一致性、服务可用性的平衡。根据分布式系统的CAP理论[1][4]，云际场景由于网络阻塞、硬件故障等造成的网络延迟丢包现象普遍存在，系统往往无法在时限内达成数据一致性（即发生了分区），因此系统必须在一致性与可用性之间做出权衡。此外，当出现新增加的存储资源时，能及时进行数据重新划分，实现负载均衡和存储服务的扩容。事实上，适合数据密集型应用的数据的存储位置和数据请求的源节点通常都是跨地理分布的，不恰当的数据分布会加剧数据访问服务性能的下降，尤其是在一个特定的事务中包含多个数据操作的情况下[5]。因此，云际管理者或数据密集型应用必须在云际资源上有效地放置数据及其副本，以保证能够灵活地提供弹性扩展能力，解决工作负载的突增和请求高峰问题。

可定制的服务质量保障 在软件定义的云际计算环境下，云服务消费者能够根据需求进行预先定义。在设计并实现一个云际存储框架时，必须全面地考虑云服务消费者预期的服务质量(QoS)指标（诸如响应时间、吞吐量、可靠性、可用性、能效等）。多个云环境的不确定性与服务性能的差异也将显著影响存储资源聚合的总体效率和可靠的数据访问服务。具体而言，在不同区域、不同时间段，每个存储节点可能存在不同的数据传输条件和模式（例如请求响应延迟、数据传输速率、最小带宽保障、能源消耗等）。我们在不同的地理位置部署了相同的客户端来调用同一数据中心的某项服务，如图1所示，请求的延迟随地理位置的不同产生了显著差异。因而，如何对不确定性进行建模与控制，如何处理请求响应性能的波动性，减少长尾延迟(tail latency)[6]是云际存储需要解决的一大难题。

安全性与可审计性 对象的易损性会受到很多因素的影响，云际环境中尤其如此。因此，数据加密、隐私保护等是加强网络安全、防止外部攻击的迫切需要。此外，如何准确有效地衡量和评估安全风险对提供一个全面的安全和风险评估模型至关重要。可审计性则是云际计算中另外一个极为重要的方面，通过跨云际的广域分布式存储，实现多源数据融合、溯源、查询等，以满足完备性与可靠性保障的审计监管的功能需求，可为云际服务的分布式记账提供数据支持。

系统容错性与高可用性 长期以来，系统容错问题一直是大规模分布式系统研究的重要课题。由

[1] Consistency：一致性；Availability：可用性；Partition tolerance：分区容错性。

于云际存储环境中不同的跨域网络条件和管理域的不确定性，容错与高可用保障技术的实现方法与机制将会变得更加复杂。当前容错的高可用性方法，例如恢复块 (recovery blocks) [1]、N-Version 编程 (NVP)[6]、检查点和回滚 (checkpointing and rolling-back) [7]、系统软状态恢复 (soft-state recovery) [8] 等手段由于运营成本高、资源消耗大、冗余开销多、系统恢复时间不确定等原因，无法直接应用于云际存储的故障处理与高可用保障机制中。冗余磁盘阵列则由于其可靠性低、数据自恢复能力弱以及对上层应用不透明而无法用于当前环境。由于重要的软件和硬件故障繁多，故障组合千变万化，如何提供不间断的可靠服务，同时降低运营成本仍然是一大难题。

云际存储的软件定义方法

软件定义方法

软件定义的基本方法将一体化、单体式应用设计进行细粒度拆分，通过硬件虚拟化为上层软件层提供应用编程接口，再在软件层对整个硬件系统进行更为灵活的管理，实现高效灵活的智能化管控。软件定义的方法实现了软件管理与硬件实现的解耦合，以及策略与机制相分离。具体而言，机制是指有机体的构造、功能及其相互关系（运行时相对不可变）；策略则指可以实现目标的方案集合（运行时可以设计）。软件定义的方法能够在虚拟化抽象的基础上，实现软件层控制与硬件的实现机制相分离，通过这种软件定义手段，能够提高策略空间设计实现的灵活性，组合硬件存储不同粒度的资源抽象；而云存储底层的存储虚拟化则无须考虑上层的设计，只须提供对应的应用编程接口，可简化软件开发流程和上层应用的实现难度。具体而言：

资源聚合度与管控细粒度化 软件定义可以实现对传统资源（诸如网络、存储等）更灵活的管理，系统的可控可管性将更细粒度化与灵活化，提供给上层应用的资源也变得更加多样、异构。

应用一体化与服务化 应用与计算框架个性化需求日益增加，为了实现对应用的透明，系统在执行层面需要对上层应用的资源需求、约束条件、定制化的偏好等属性进行准确细致的描述。因此，软件定义系统既要能够管理与调度大规模的网络化

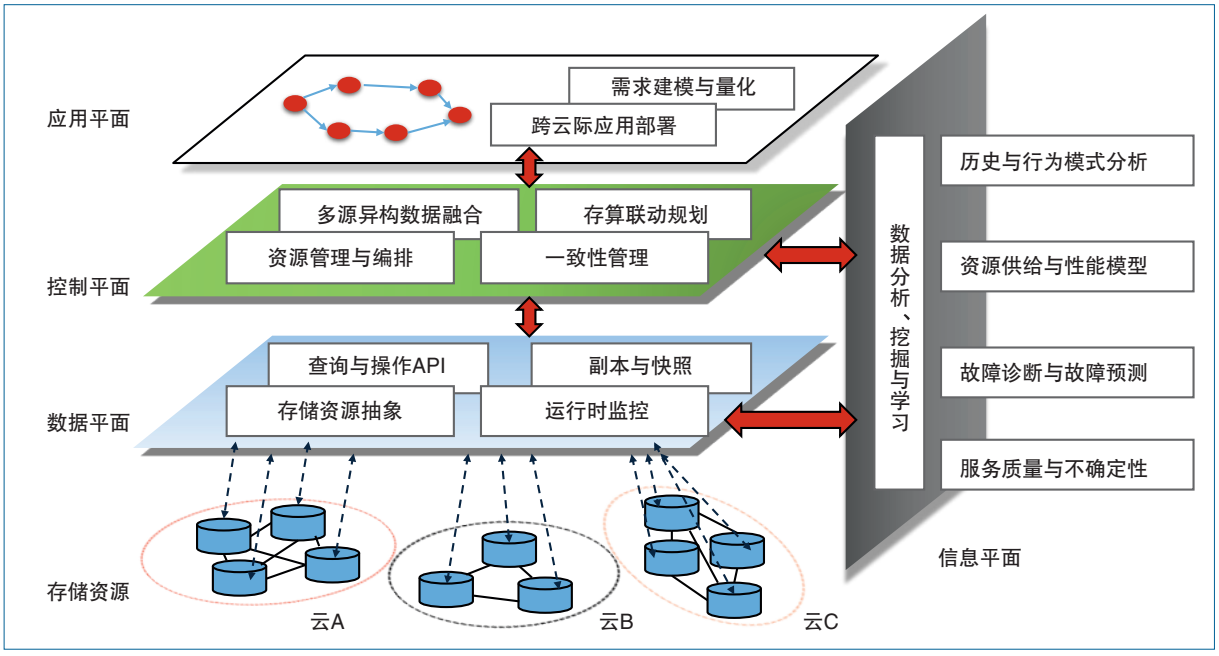

图2 云际存储的软件定义架构

资源（计算、存储、网络、平台），同时也要支撑这些应用与框架，同时以服务化、可编程的形式提供服务。

多源的信息、数据高度融合化 软件定义使得传统云计算中基础设施的各种软硬件资源能够通过统一的虚拟化软件层加以管理，与传统的资源管理系统相比，软件定义的云计算系统可以获得更多细粒度的信息与运行数据，例如应用层特征信息和系统层特征信息等。这些多源信息有助于对负载行为、系统的运行状态进行判别与预测。利用数据挖掘与机器学习方法可以使得资源的管控更智能，为调度决策提供更精确的信息，同时也能实现前瞻性的故障防控诊断、集群自动可扩展伸缩等。

软件定义的云际存储

相比于传统的云存储，云服务提供者之间无论在设备结构、硬件组成方面，还是在平台可用性、可靠性、能耗、隐私性、应用适配性、价格等多属性方面所存在的异构性将进一步加大，这种差异可以通过软件定义的软硬件解耦合的手段进行屏蔽，为上层存储资源的聚合与统一调度的平台提供技术支持。

通过软件定义的云际存储架构与设计，用户能够根据应用策略来配置、组合和使用存储服务，并将它们部署在一系列云际环境中的多种硬件资源上。

在服务质量控制方面，相关的数据访问和存储服务将从硬件资源池中分离出来，形成统一软件控制层。该控制层可以根据服务等级协议 (Service-Level Agreement, SLA) 以及消费者需求，灵活组合已有的存储资源提供存储服务，通过冗余的组件单元、数据副本来控制并降低网络服务质量不确定性所带来的影响。

在互操作性方面，软件定义的控制器将起到决定性作用，它能够通过统一的消息通信协议和消息总线对异构的接口加以整合，来消除云际间数据传输、迁移、共享中的各种障碍，使得异构的接口与协议的组件能够灵活地进行适配。

在弹性容量与可靠性方面，云服务消费者可根据上层应用动态的需求，通过软件控制层动态调整资源与配置。例如，当负载波动出现激增时灵活扩容，或根据实时价格动态调整服务水平，为高风险组件增加备份数量、提高检查点频率，保证服务性能与云服务消费者的服务质量，对系统可靠性、安全性进行统一管控。

云际存储中的关键问题

为实现广域网环境下多云之间协作的存储服务，首先我们需要创建跨地理分布式的存储资源池，将云中所有物理存储设备转化为统一、虚拟、共享的存储资源池中的基本单元。其次，把存储的控制与管理从物理设备中抽象与分离，将其纳入统一的集中化管理中，形成硬件与上层应用之间的软件层。

如图 2 所示，我们按照软件定义的思想对系统进行了平面化设计，存储资源经过虚拟化实现了统一的资源抽象；通过控制平面与数据平面解耦合，实现了管控与运行的分离；同时辅以信息平面提供智能的数据挖掘与分析能力，提高了软件定义的云际存储的服务质量，进而为应用平面的应用部署与运行提供了强有力的支持。与此同时，由于数据是一种特殊的存储物质，我们将把数据设备从存储设备中独立出来，使得跨存储设备的数据服务成为可能。数据即服务 (Data as a Service, DaaS) [9] 将为云际计算的其他模块（例如云际资源管理与任务调度、云际网络管理等）提供更为便捷、高效的基础设施。

云际存储的可表达：抽象与需求

存储资源抽象与需求模型 为了实现应用对云际资源需求的精确评估，进而高效地利用云际间的资源，提高资源共享效率，特别是针对云际存储能力异构、地理分布等因素，我们研究云际环境下不同地理位置或服务商的云存储的灰盒资源抽象模型，对云际间存储资源的价格、一致性、可靠性、服务等级协议等一系列属性建模，为云际数据处理与存储管理提供支持；我们提出了一种可扩展、声明式的建模机制来实现高层抽象、云服务消费者个

性化需求与云际资源特征的有机融合，给出基于大数据复杂性、可计算性理论的软件定义云际存储计算效率的表示与分析方法，能够权衡精度、时效性与任务执行效率之间的关系。此外，我们针对云际计算环境中跨地理位置分布所造成的网络服务质量、请求响应等不确定性问题，通过数据驱动的方法定量建立了资源编排与分配过程中的系统不确定性模型。

系统性能追踪与分析 针对大规模云际分布式系统结构拓扑、调用链依赖关系复杂的问题，深入研究基于性能模型的云际资源评估与优化选择，提出了基于分布式追踪的组件交互关系捕获机制，实现细粒度、组件级的性能模型自动构造，建立应用性能与云际资源供给的关联关系，并综合考虑组件关联及交互代价、资源依赖及定价等因素，实现了云际资源的量化评估与优化选择。针对云际大数据计算系统资源规模化与动态扩展变化等的特征，设计了一种可扩展的大数据系统资源运行监控机制。

云际存储的可管理：高效协同

除了需求的抽象与性能驱动的系统建模之外，我们还将以云际计算环境跨地理位置分布的数据资源为核心，针对云际计算设施跨地域、管理自治性的特征，解决在云际存储节点上进行数据放置、数据存取，实现基于数据位置感知与性能感知的应用部署与调度等高效协同问题，实现面向云际大数据存储与高效处理的软件定义架构、跨云际数据中心的数据服务部署、大规模并发访问等重要功能，实现全局的效能优化与多副本开销最小化，降低云际不确定性对服务性能带来的影响。整体的系统架构如图 3 所示，主要包括可扩展的名字空间管理、元数据管理与协同、数据放置与副本管理、面向数据

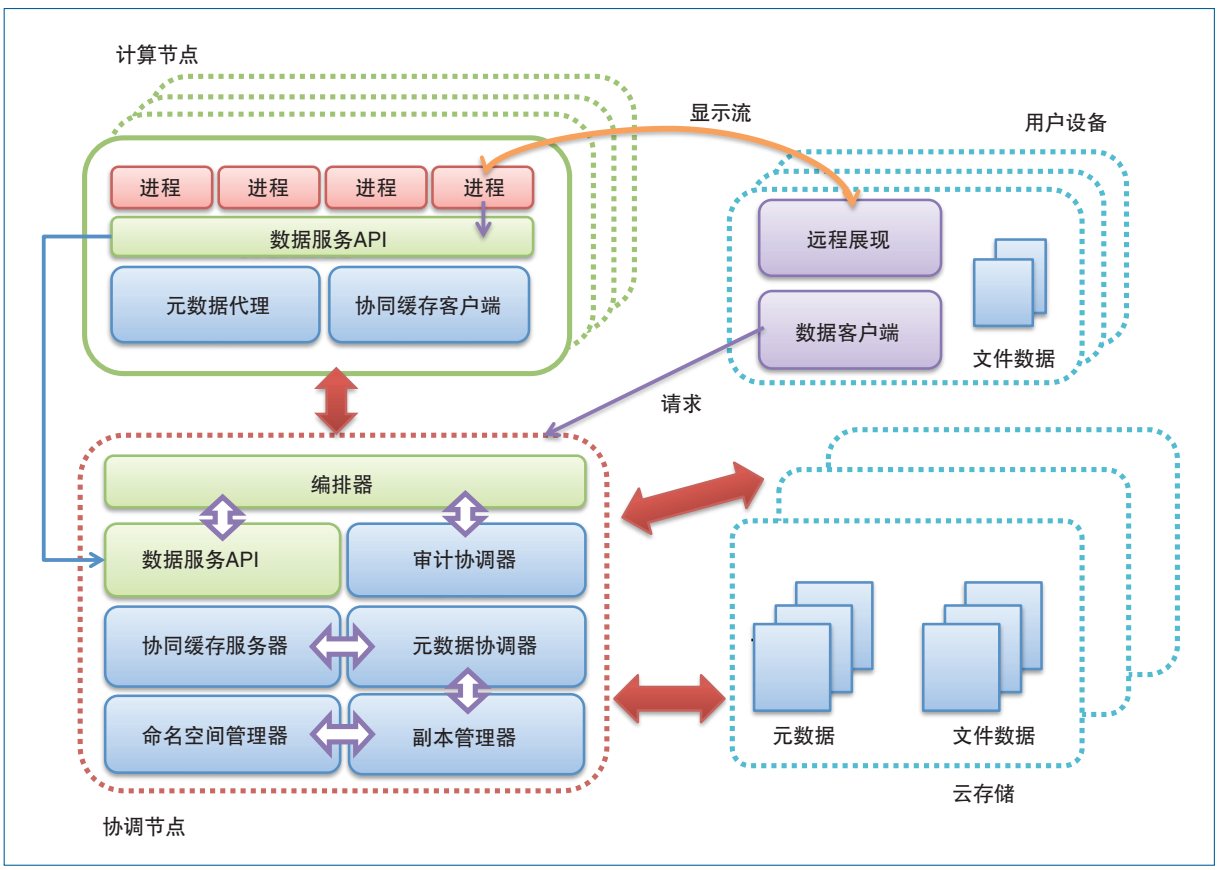

图3 云际存储系统设计

服务的协同缓存等内容。

数据放置与编排 大量数据密集型应用时，往往在一个特定的事务中又包含多个数据操作[5]，而传统基于哈希的负载均衡的存储模式无法高效应对上述情形。因此，全局动态调整数据位置(data placement)，整体减少系统通信开销与网络流量，提高请求响应速度和效率尤为重要。为此，我们需要研究数据项或文件块之间的相关关系，将强相关的数据聚合到相同或相近的位置；同时研究请求模式与用户组之间的关联关系，实现数据项的动态划分与调整。

冗余与副本管理 端到端延迟(end to end latency, e2e time)对于云际存储服务至关重要，广泛使用的多副本技术可以减少上述的服务请求长尾[1,7,8]。但是针对云际服务和资源分布的特点，我们仍须进一步研究多尺度的副本冗余机制，例如存储服务内部的多个请求处理组件冗余化，以便提高并行处理能力、数据中心内服务请求的关键路径优化、请求分发到部署在云际环境中的多份服务副本等，实现横向扩展与垂直的多尺度整合。根据不用的应用特点自适应地选择对应尺度的策略，实现负载均衡缓解请求长尾现象发生。

可定制的一致性配置 云际计算广域网环境中数据多副本间的同步化问题将变得更为突出，我们需要研究应用数据存储的一致性模型与副本策略，针对需求提供不同粒度高效自适应的一致性控制方法，减少PUT操作的系统开销。此外需要提供定制化的接口支持管理功能可编程，解决访问时延、网络流量消耗、能耗、高可靠性等多目标优化问题。

数据服务的效率与可靠性 为了实现数据即服务的服务质量，我们可以采用跨云际的内存数据缓存方法，来降低GET操作的长尾延迟问题；针对一些软件即服务应用，通过预取技术(prefetching)提前将数据块按需载入，实现从构建到运行的全生命周期管理；设计并实现用户透明的故障恢复机制，解决系统中的单点问题[10,11]，为云际计算环境提供高可靠、低延迟、大规模的数据存储、数据存取与共享等能力。

云际存储的可计算：计算效率

在大规模的云际存储系统中往往涉及大量的计算需求，这也对计算过程中的实时性、准确性、资源消耗等系统效率提出了更高的要求。

高效的优化求解框架 正如优化问题的解空间会呈现指数型增长的趋势一样，调度和编排问题的求解复杂度也随着规模的增大而急速增加。例如，在所有可用候选配置空间中挑选出请求延迟最小的方案，利用基于图的动态划分来决定关联数据的放置位置等优化问题，在大规模拓扑结构或候选集的情况下通常非常耗时。另外，应用对数据存储需求的动态变化，约束条件增多，定制化属性多元化等，都将进一步增加计算和处理的复杂度，因此，我们需要利用分布式架构对相关算法和计算框架进行并行化处理，以加速上述问题的求解。除此以外，统计分析、机器学习等方法有助于利用性能日志(tracelog)和相关历史数据来实现资源利用率、故障发生、长尾请求和慢任务(straggler)等系统行为的预测，可大大减少解空间的无用搜索，加速最优化的求解过程。

增量与近似计算 根据云际环境中系统动态性与存储容量的弹性需求，我们需要针对大规模调度对强实时性需求与系统快速动态演化的特征，来研究基于机器学习的在线迭代的资源与数据编排机制，以实现动态增量的重调度，达到快速迭代的应用组件编排、资源分配，保障运行时的调度质量与吞吐量。此外，我们还需要研究云际计算环境分布的多源异构数据融合技术，通过适当降低计算对精确性的要求，满足大数据应用在数据量、精度和时效性方面的权衡；针对大数据计算增量性和近似性特点，分析大数据计算体系中存储与计算、网络等资源分配的联动关系，利用超卖、动态迁移等技术支持动态及非精确的资源优化分配策略。

能效优化的就近计算 存算联动的系统优化有助于数据传输的最小化，以减少计算的总时间。类似于数据本地性(data locality)和近端处理原则的基本思想，我们提出了一种移动数据和移动任务

相结合的混合模型。该模型能够考虑数据位置、任务的依赖关系、共享数据的亲和度等，以此来缓解不均衡的数据通信与传输。进一步地，站在更加广泛的智能城市和物联网的计算环境角度，考虑到大量的数据移动与传输过程中所产生的能源消耗，我们提出了一种全新的“**数据燃料 (using data as fuel)**”的能效优化方法，即将数据传输所产生的热量进行二次利用，实现供热、家居供暖等功能，减少能源的浪费。为此我们将对云际环境中所有的能量生产者与消费者进行全局规划建模，考虑诸如数据存储位置、副本部署、计算任务、数据传输以及智能城市中传感设备、热力站、智能楼宇等实体，形成一个全域、绿色的云际计算环境。

结语

软件定义的云际计算的出现，可以解决全时全域的优质服务和灵活多样的个性化定制的需求，是新一代云计算模式。软件定义的云际存储作为云际计算中的重要组成部分，将在数据存储、大数据处理、大规模数据服务支撑中起到至关重要的作用。我们相信，对上述问题的深入研究与实践必将提升云际存储资源的聚合效率，降低云际不确定性给服务性能带来的影响，实现软件定义的云际大数据高效能存储管理，显著提高系统可靠性，实现低成本、低延迟数据存储与共享等能力。■

沃天宇

CCF专业会员。北京航空航天大学计算机学院副教授。主要研究方向为系统虚拟化、大型分布式系统、时空数据处理与挖掘等。

woty@act.buaa.edu.cn

杨任宇

北京航空航天大学大数据与脑机智能高精尖创新中心博士后研究员。主要研究方向为大规模分布式系统、资源调度与管理、系统可靠性与容错等。

renyu.yang@buaa.edu.cn

王媛冬

北京航空航天大学计算机学院博士生。主要研究方向为数据挖掘、时空数据分析等。

wangyd@act.buaa.edu.cn

其他作者：胡春明　徐　洁

参考文献

[1] "Internet Plus" Strategy. http://www.usito.org/news/china-pursues-internet-plus-strategy.

[2] 梅宏, 黄罡, 曹东刚, 等. 从软件研究者的视角认识软件定义[J].中国计算机学会通讯, 2015,11(1): 68-71.

[3] Häyrinen K, Saranto K, Nykänen P. Definition, structure, content, use and impacts of electronic health records: a review of the research literature[J]. *International Journal of Medical Informatics*, 2008, 77(5):291.

[4] Brewer E. CAP twelve years later: How the "rules" have changed[J]. *IEEE Computer*, 2012, 45(2):23-29.

[5] Quamar A, Kumar K A, Deshpande A. SWORD: scalable workload-aware data placement for transactional workloads[C]//*EDBT '13 Proceedings of the 16th International Conference on Extending Database Technology*. 2013:430-441.

[6] Dean J, Barroso L A. The tail at scale[J]. *Communications of the ACM*, 2013, 56(2):74-80.

[7] Vulimiri A, Godfrey P B, Mittal R, and et al.Low latency via redundancy[C]// *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. New York: ACM Press, 2013:283-294.

[8] Wu Z, Yu C, Madhyastha H V. CosTLO: cost-effective redundancy for lower latency variance on cloud storage services[C]//*NSDI'15 Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2015:543-557.

[9] Zheng Z, Zhu J, Lyu M R. Service-Generated Big Data and Big Data-as-a-Service: An Overview[C]// *Proceedings of IEEE International Congress on Big Data*. IEEE, 2013:403-410.

[10] Yang R, Zhang Y, Garraghan P, et al. Reliable computing service in massive-scale systems through rapid low-cost failover[J]. *IEEE Transactions on Services Computing*, 2016.

[11] Yang R, Wo T, Hu C, et al. D2PS: a dependable data provisioning service in multi-tenants cloud environments[C]// *Proceedings of the 17th IEEE HASE*. IEEE, 2016: 252-259.