doi:10.3772/j.issn.1002-0470.2020.01.001

松耦合数据中心体系架构研究综述①

赵博彦②[*,**,***] 侯 锐③[*] 张乾龙[**,***] 包云岗[**] 张立新[**] 孟 丹[*]

([*]中国科学院信息工程研究所 北京 100093)

([**]中国科学院计算技术研究所 北京 100190)

([***]中国科学院大学 北京 100049)

摘 要 数据中心作为信息产业的基础近年来得到了飞速的发展,同时也面临着大数据时代带来的新挑战。传统数据中心架构与数据中心新兴应用特性的不匹配越发明显,同时数据中心的资源利用率也持续偏低,然而数据中心的成本和功耗却在不断走高。面对这些问题,松耦合数据中心架构的发展成为数据中心重要的发展趋势。将数据中心资源松耦合化,可以突破传统物理界限,在提高资源利用率的同时加强成本控制。本文首先总结了松耦合数据中心发展现状,然后提炼了松耦合数据中心架构的关键技术,最后对该领域未来的发展前景进行了展望。

关键词 数据中心;松耦合;通信机制;资源共享;资源分配;数据备份;计算机体系架构;可编程门阵列(FPGA)

0 引 言

大数据时代的到来带动了数据中心的飞速发展,近年来谷歌、微软、亚马逊、阿里巴巴等企业在全球范围内掀起了建立大规模数据中心的热潮。相关数据表明,2016 年全球数据中心市场规模已达到 451.9 亿美元[1]。

现在的主流数据中心配置构建方案通常选择主流中高端处理器平台,并采用集群架构作为整体架构。数据中心服务器系统的设计要在追求性能的同时控制成本和能耗。然而,大数据的兴起为数据中心设计提出了新的挑战。传统数据中心面临的挑战集中在4个方面:(1)传统数据中心日益严重的过分定制导致成本居高不下;(2)新兴的数据中心应用需求与传统数据中心平衡设计不匹配;(3)数据中心普遍面临资源利用率较低的问题;(4)现有数据中心架构很难动态使用空闲资源。

传统高端处理器的设计主要侧重于如何改善单节点处理器芯片的计算性能和效率。现有的数据中心大多采用商用 X86 处理器搭建,如 Intel 的 Xeon 系列处理器。人们针对桌面应用和高性能应用对这些通用处理器的访存性能做了大量的优化,例如增加访存总线的位宽、提高前端总线的频率、优化访存调度算法等。这些优化技术极大地提高了传统应用的访存性能。然而,相关研究表明,新兴的数据中心应用与传统的桌面应用和高性能应用有着截然不同的特征[2]。数据中心的典型应用往往需要多个计算节点之间的大规模协同工作,并且表现出不同的资源需求,与现有服务器中的系统平衡设计并不匹配。上述应用包括网络搜索、MapReduce 的数据分析、社交媒体的分布式内存缓存和桌面云应用等[3]。这些应用强调的是内存的容量,而非带宽。它们的内存占用量(footprint)很大,但是内存带宽的利用率却比较低,而且发往内存控制器的请求序列

① 中国科学院前沿科学重点研究项目(QYZDB-SSW-JSC010)和国家自然科学基金(61522212)资助项目。

② 男,1990 年生,博士生;研究方向:计算机体系结构,数据中心架构;联系人,E-mail:zhaoboyan@iie.ac.cn

③ 通信作者,E-mail:hourui@iie.ac.cn

(收稿日期:2019-02-15)

呈现出稀疏和不规则访问特性[3-8]。这类型的应用运行在传统的 X86 处理器上,浪费了大量的访存带宽和功耗。因此,设计数据中心服务器的处理器芯片,需要充分从系统的角度去考虑单节点处理器芯片结构的优化,增加对多节点协同工作的支持,从而实现一个更加高效、低成本的数据中心系统。

数据中心不同于传统的企业机房,只需要提供单一或几项应用,如邮件系统、办公系统等。数据中心承载着多种应用,且应用种类越来越多样化,并且集约化的大型数据中心会越来越多。为应对多样化负载以及出于可扩展性的考虑,数据中心管理员通常倾向于依据峰值需求简单地为每台机器配置资源,日益严重的过分定制(over-provision)导致数据中心成本始终居高不下,并且这种趋势愈演愈烈。研究人员针对数据中心大数据应用进行了大量的评估,发现 Hadoop 应用的内存总线带宽利用率平均只能达到 15%,Spark 应用的内存总线带宽利用率平均为 40%[9]。Ren 等人[10]对淘宝的 Yunti 数据中心进行了评估,发现其内存容量利用率平均在 30% 左右。如此高的资源闲置比例,导致了大量的成本和功耗的浪费。如何从数据中心整体的角度,根据资源平均利用率来配置每台机器的资源,是一个值得深入研究的问题。

目前数据中心的服务器节点都配备了独立的内存、处理器、磁盘及其他资源。即便数据中心服务器每个节点的利用率都不尽相同(甚至相差很远),当前架构很难支持管理员能够动态使用远程空闲资源。对于远程内存的借用,前人基于以太网和 Infiniband[11]做过一些尝试[12-15],也获得了一些性能上的提升。然而,这些研究都没有从本质上解决远程资源的高效动态借用。基于以太网和 Infiniband 的远程内存借用只能通过远程直接数据存储(remote direct memory access, RDMA)方式,不能通过 Load/Store 指令直接访问。而且以太网和 Infiniband 需要额外的交换机、适配器,以及协议栈到 PCI-e 协议栈之间的转换(例如以太网需要 TCP/IP 到 PCI-e 协议的转换),访问路径长,效率较低。因此,设计一套专门针对数据中心节点间资源借用的定制网络协议具有非常重要而现实的意义。

将数据中心的资源进行松耦合化是一个行之有效的方法,代表了数据中心未来的重要发展趋势。该架构突破了传统服务器的物理界限,允许任意节点能够根据负载需要动态地以“借用”的形式访问远程节点的空闲处理器、内存以及各种外设资源。这种资源共享能够有效避免目前普遍存在的过分定制的问题,从而实现成本有效的数据中心服务器架构。

本文从松耦合数据中心的工业界和学术界研究现状出发,分析总结松耦合数据中心的主要特征,希望对松耦合数据中心的研究有所帮助。

本文的结构如下:第 1 节对工业界松耦合数据中心研究现状进行概述;第 2 节详细介绍当前学术界对松耦合数据中心的研究现状;第 3 节对松耦合数据中心架构特点进行总结,讨论设计松耦合数据中心架构时的关键技术;根据第 3 节的分析,第 4 节探讨了松耦合数据中心架构的研究方向并进行了总结。

1 工业界研究现状

工业界对松耦合数据中心的研究基于不断的产品迭代,针对企业自身面临的特殊问题,在追求稳定的同时积极提高资源利用率以达到控制成本的目的。一些互联网企业如 Google 和 Yahoo 等数据中心纷纷弃用高端商用高性能处理器,选择廉价机器机群加上大规模数据处理软件(如 hadoop)来运行自己的业务。工业界对松耦合数据中心架构的研究往往考虑了很多兼容性和自适应的问题,无法涵盖数据中心复杂的应用场景。

1.1 Intel 公司的 Rack Scale Architecture

Intel 公司在 2013 年 10 月正式公布了数据中心新一代机柜式架构设计理念(rack scale architecture, RSA),该架构基于片上光互联技术将机柜内的服务器、网络、存储等部件整合,变成一个计算池、网络池、存储池的概念[16]。整个机柜可以根据应用的需求动态地为每个节点分配不同的资源。从而,降低数据中心的总体持有成本(total cost of ownership, TCO)、提高服务器平台的灵活性、提高数据中心的

数据处理能力和硬件资源利用率。RSA 架构[17]如图 1 所示,RSA 包含如下 5 个关键技术:(1)参考体系结构(reference architecture)和软件调度层(orchestration software);(2)Intel 开放网络平台(ONP:open network platform);(3)PCIe 固态硬盘(solid state disk,SSD)和 caching 等存储技术;(4)光纤交换结构(photonics and switch fabrics);(5)支持多种类型的处理器和加速器(例如 Atom、Xeon 和 Quark 等)。在将 RSA 中的这些技术应用到云服务中后,Intel 公司评估的结果是:在使用了硅光技术后,电缆线的需求量降低了 3 倍;网络上行和下行速度都提高了 2.5 倍;服务器密度(服务器/机柜的比例)增加了 1.5 倍;能耗降低了 6 倍。采用了 RSA 的服务器将为公共云、私有云、大数据处理提供更加便捷高效的服务。

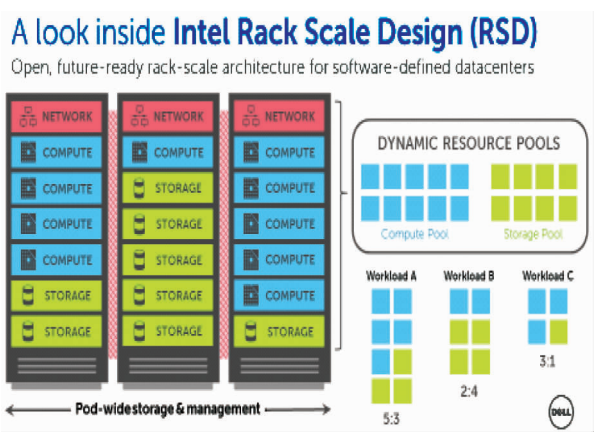

图 1　RSA 架构示意图

1.2　微软公司的 Catapult 项目

为了提高现有商用数据中心的处理能力,微软设计并构建了一种可组合、可重配置的网络互连技术——Catapult[18],来加速大规模软件服务的部分功能。该设计把现场可编程门阵列(field-programmable gate array,FPGA)作为一种细粒度的加速器,将部分软件的工作交由 FPGA 来处理,从而减轻了处理器的负载,可以加速数据中心里的大规模服务应用。Catapult 的网络架构是一个 6×8 的 2 维 torus 网络,每个网络节点是一个由中等尺寸 FPGA 和本地 DRAM 组成的扩展板。该互连模块被嵌入到 48 个服务器中,每个服务器上插有一个扩展板。该设计允许将 FPGA 分成多个组,从而将任务按组分配到 FPGA 来完成。微软构建了一个中等规模的 Catapult 互连网络,包含 1 632 台服务器,测试了它在加速必应搜索(Bing)引擎的效果。具体加速方案是,将必应搜索引擎的排名系统(ranking stack)中的部分软件功能(打分功能)交由一个 FPGA 组来实现。Catapult 项目 FPGA 工作流程如图 2 所示。该 FPGA 组由 8 个 FPGA 构成。当一台服务器为一个文档进行打分的时候,该服务器首先对文档进行格式转换,进而将转换后的文档注入到本地的 FPGA。然后,该文档通过 FPGA 网络路由到负责打分的 FPGA 组,经过 8 级的 FPGA 流水线得到最终打分结果。该打分结果再经由 FPGA 网络路由回请求节点。评估结果表明,与纯软件实现方案相比,Catapult 互连技术在维持同等延迟分布的情况下,每个 ranking 服务器的吞吐量提高了 95%,或者在维持同等吞吐量的情况下,延迟减少了 29%。

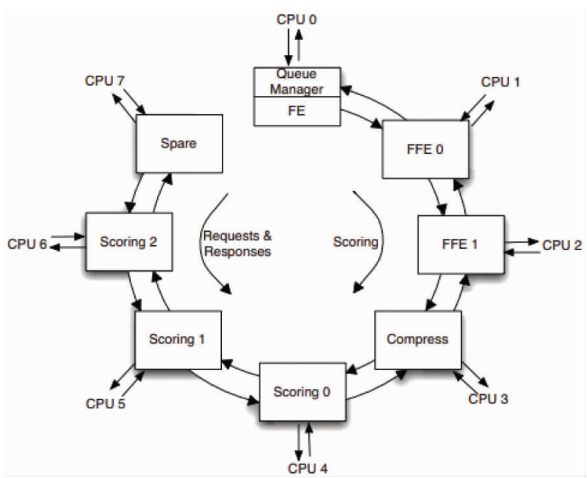

图 2　Catapult 项目 FPGA 工作流程[18]

Catapult 项目成功实现了加速器资源的松耦合化,证明了松耦合数据中心架构的可行性和前瞻性。服务器的计算压力由 FPGA 加速器分担后,可以减轻服务器的过分定制问题,降低成本的同时也提高了数据中心的整体利用率。

微软 Catapult 项目组成功实现加速器资源松耦合架构后,继续深入开发。在 2015 年先后在 FPGA 上实现了高带宽的无损数据压缩[19]和深度卷积神经网络加速[20]。相比于使用图形处理器(graphics processing unit, GPU)的加速方案,使用 FPGA 加速

方案的功耗只需要 11%，整体吞吐量可以达到 63%。Catapult 项目组在 2016 年提出了一种云计算规模的加速架构，该架构在网络交换机和服务器之间部署了由 FPGA 组成的可重构逻辑层，使用可重构逻辑加速网络平面功能和应用性能[21]。在 2018 年先后发表 2 篇关于深度神经网络（deep neural network，DNN）的加速工作，一篇使用 FPGA 加速 DNN 推理运算[22]，一篇使用专用的神经处理单元（neural processing unit，NPU）加速 DNN 运算[23]。针对微软云计算平台 Azure 的网络协议栈提出了加速网络 AcceINet，使用基于 FPGA 的自定义智能网卡（Azure SmartNIC）网络协议栈从服务器卸载到网卡上，可以提供小于 15 μs 的虚拟机端到虚拟机端的 TCP 访问延迟和 32 Gbps 的吞吐量[24]。

1.3　Open Compute Project

Open Compute Project（开放计算项目）最早由 Facebook 公司联合 Intel 公司、Rackspace 公司、Goldman Sachs 集团（高盛集团）和 Andy Bechtolsheim 在 2011 年发起[25]。最初只是公开分析 Facebook 公司数据中心产品设计，现在已经有 IBM、英特尔、谷歌、微软、戴尔、思科、诺基亚、联想、阿里巴巴等众多企业加入该计划中，共同参与数据中心的设计和分享。该项目旨在设计、使用和实现高效的可扩展计算，追求更佳的技术创新和更低的技术复杂性。加入该项目的个人和组织都可以与他人共享知识产权，共同推动数据中心领域的不断发展。

整个项目包含了数据中心建设的全部内容，从基础建设到上层架构，拆分为 10 个子项目以及 5 个区域项目。这 10 个子项目为数据中心设施、硬件管理、高性能计算、数据中心网络、机柜和电源、服务器、存储、电信产业、系统固件（筹备中）、安全（筹备中）。5 个区域项目包括中国、欧洲和日本等 5 个国家和地区的数据中心工程[25]。

开放计算项目也在进行数据中心松耦合化的探索工作，例如对软件定义 SSD 的研究项目可以实现 2 种存储松耦合的实现方式。

1.4　华为 HTC-DC

华为公司在 2014 年提出了高通量计算数据中心（high throughput computing data center architecture，HTC-DC）的概念，并作为下一代数据中心（DC 3.0）的研究方向。HTC-DC 架构（见图 3）在以虚拟化为主体的第 2 代数据中心的基础上，通过资源松耦合化和统一的互联方式，达到 PB 级的数据处理能力，同时还具备了更好的可扩展性和能耗效率[26]。

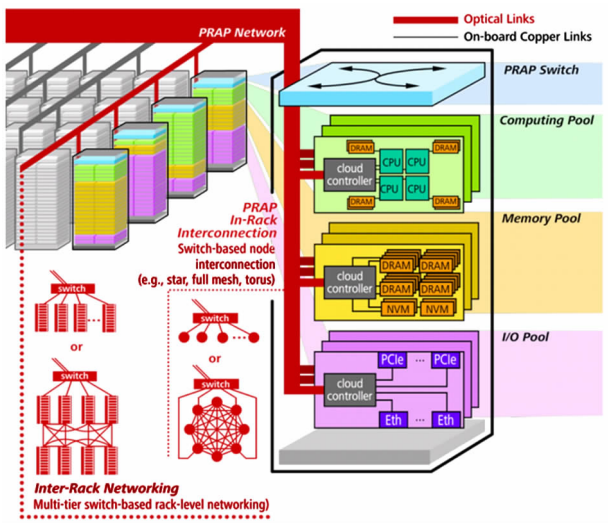

图 3　华为 HTC-DC 架构[26]

高通量计算数据中心架构将计算资源、内存资源和 I/O 资源划分为不同的资源池，每个池内是对应的服务器节点。资源池内以及资源池之间通过专用的互联协议连接，资源的分配和管理由专用的数据中心操作系统完成。数据中心操作系统根据不同应用的资源需求从资源池中划分相应的资源，还可以根据应用的需求变化动态调整可供使用的资源数量。

在 2018 年德国汉诺威国际信息及通信技术博览会（CEBIT）上，华为公司发布了基于人工智能的数据中心技术方案[27]。采用人工智能优化运行算法，实现数据中心基础设施整体功能的智能化融合，为数据中心的松耦合化打下坚实的硬件基础。

1.5　Gen-Z

Gen-Z 标准发布于 2016 年，现已有超过 60 余家公司和机构加入该标准，包括 AMD、谷歌、戴尔、ARM 和华为等众多知名企业[28]。

Gen-Z 标准是一种新的数据访问技术，可以在直连或者互联等网络拓扑结构上提供低延迟的、内

存语义级的数据和设备访问。Gen-Z 结构组件利用内存语义通信以最小的开销在不同组件上的存储器之间移动数据。该技术可以实现低延迟的数据直接读写操作,并且保证应用程序和操作系统尽可能少地参与其中,可以在不牺牲灵活性的前提下提供最高的性能。更关键的一点是,这些操作可以无需修改操作系统和应用程序中间件。该技术还支持缓存一致性、原子操作、PCI 以及 PCI-e 总线技术。

在该技术的支持下,内存的松耦合化可以以更灵活更兼容的方式实现。兼容 PCI 和 PCI-e 总线技术的 Gen-Z 技术也可以有效地支持远程输入输出(input/output,I/O)设备的访问,实现 I/O 设备的松耦合化。

1.6 远程 GPU 访问架构 rCUDA

rCUDA 是 Remote CUDA 的缩写,是一种远程 GPU 虚拟化软件架构,可以为 GPU 集群提供全虚拟化支持,进而提高整体性能[29]。当一个没有配置 GPU 的物理节点需要使用 GPU 资源加速计算时,一个或者多个远程 GPU 会被分配给该节点使用。存储在本地内存中的数据和程序会迁移到远程 GPU 的显存上,程序执行内核也会在相应的远程 GPU 上启动运行。rCUDA 架构是一种 Client/Server 结构。Server 端需要监听 TCP 端口,接收服务请求。Client 端在 CUDA 架构应用程序编程接口(application programming interface,API)的基础上封装服务请求并发送给 Server 端,由 Server 端将服务请求解析并在 GPU 设备上执行。rCUDA 架构可以通过 GPU 虚拟化,减少 GPU 集群内的设备数量,达到降低成本、节约能源和维护费用的目的。rCUDA 架构可以和 CUDA 的编程接口完全兼容。

1.7 分布式共享内存系统 DSM

分布式共享内存系统(distributed shared memory system,DSM)对系统内所有节点的内存进行全局统一编址,系统内的所有内存都可以被全局共享。DSM 可以分为软件 DSM 和硬件 DSM 两类,DSM 架构如图 4 所示。

软件 DSM 系统的第一个原型是李凯教授于 20 世纪 80 年代提出的 IVY 系统[30],发展至今已经实现了各种软件 DSM 系统,包括美国莱斯大学的 TreadMarks[31]、DEC 公司的 Shasta 系统[32]、卡内基梅隆大学的 Midway 系统[33]、马里兰大学的 CVM[34]和中国科学院计算所的 JIAJIA[35]等。软件 DSM 系统通过软件维护整体的内存一致性,内存共享在消息传递系统上实现。软件 DSM 系统相比硬件 DSM 系统硬件实现简单,且支持共享内存系统编程接口。但是通过软件维护一致性协议也带来了较大的开销,且很难做到用户透明和软件兼容。

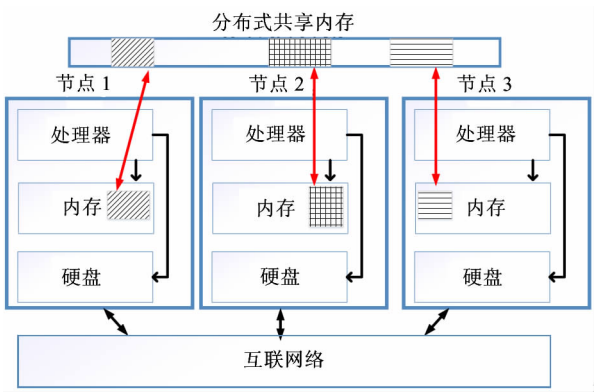

图 4 分布式共享内存架构示意图

硬件 DSM 主要包括麻省理工学院的 Alewife Machine[36],SGI 公司的 Origin 系统[37],ScaleMP 公司的 vSMP[38]以及斯坦福大学的 DASH 系统[39]。硬件 DSM 系统相比于软件 DSM 系统,具有低延迟和可扩展性高的特点。硬件 DSM 系统主要面向高性能计算领域,较高的成本是难以普及的重要原因,也不符合数据中心成本有效的要求。

2 学术界研究现状

2.1 松耦合内存

密歇根大学的 Lim 等人[40]提出了松耦合内存(disaggregated memory)的概念。为了解决单节点内存容量不足以及内存利用率低导致的节点内存过度配置的问题,实现灵活以及低成本的内存资源池,他们设计了一个单独的大容量内存节点作为可供灵活分配的内存资源池。该节点集成了大量的内存控制器,可以支持大容量的内存。内存节点与计算节点通过 PCI-e 或者超传输总线(hypertransport,HT)等互连,内存节点的内存可以被计算节点共享,不支持

计算节点间共享内存的一致性。他们基于虚拟机管理器(hypervisor)实现了基于页面粒度的远程内存管理,远程内存对上层的虚拟机操作系统完全透明。整个系统的设计和实验均基于一个 Trace 模拟器进行评估,Trace 收集自真实的数据中心和另一个周期精确的模拟器,实验结果表明松耦合内存技术可以有效提高单节点的内存容量,提高应用的性能。图 5为松耦合内存刀片服务器示意图。通过成本和功耗分析发现,该系统可有效提高性能价格比(performance-per-dollar)。然而该结构的缺点是:(1)内存节点的配置不够灵活,在机柜内如何配置内存节点与计算节点的比例与应用密切相关;(2)整个系统的内存访问带宽受限于内存节点的出口带宽,当多个节点同时运行高带宽应用时,内存节点的互连接口会成为系统的性能瓶颈;(3)HT 互连设计复杂且不具备通用性,PCI-e 互连不支持 load/store 直接访问,而且需要额外的 PCIe switch 芯片实现多个计算节点的扩展。

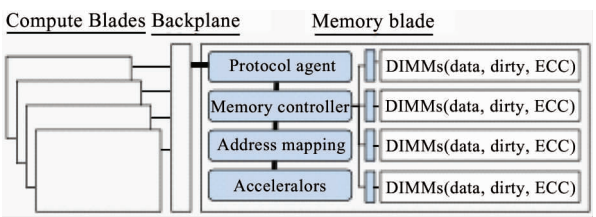

图 5　松耦合内存刀片服务器示意图[40]

2.2　Scale-out NUMA 存储架构

Scale-Out NUMA(soNUMA)[41]是一种低延时的分布式存储架构,专门为数据中心的分布和可扩展应用(scale-out application)而设计。其架构如图 6 所示,soNUMA 具有以下特点:(1)用精简的内存网状连接结构替代多层的网络栈结构;(2)通过与 RDMA 相似的远程内存操作方式来支持全局分区虚拟地址空间访问,使得它避免了维护全系统的一致性问题;(3)用廉价的 cache-to-cache 传输替代时延较长的 PCI-e 总线传输,减少访问延时;(4)优化了机架规模的部署。soNUMA 通过一种无状态的消息传递协议,直接在 NUMA fabric 上附加了一层 RDMA 风格的编程模型。为了便于应用程序、OS 和 NUMA fabric 之间的相互协作,soNUMA 在每个节点内集成了远程内存控制器,安全地对应用暴露了全局地址空间。这使得 soNUMA 在结构相对简单的同时获得了与 RDMA 相近的远程内存访问能力。soNUMA 采用了 2 种简单的方法来减少延迟。第 1 种方法是使用一种运行在 NUMA 内存结构上的无状态请求/应答协议。该协议可以极大减少或消除在网络栈、复杂网络接口和跳转上产生的延时。第 2 种方法是在节点局部连贯层集成协议控制器,从而避免了在速度缓慢的 PCIe 的接口上进行状态复制和移动数据。

基于周期精确的全系统仿真结果表明,soNUMA 执行远程读操作的延时在本地 DRAM 延时的 4 倍以内,它可以充分利用可用的内存带宽,并能达到每处理器 10 MB/s 的远程内存操作。总的来说,soNUMA 结合了 CC-NUMA 和 RDMA 的优势,同时较为成功地避开了它们各自的缺陷。然而 soNUMA 存在程序兼容性的缺点,现有应用需要修改源码,调用相应的函数库才能充分利用 soNUMA 的特性。此外,soNUMA 只支持基于通信原语(send/receive)的通信,不支持远程内存的 load/store 直接访问。

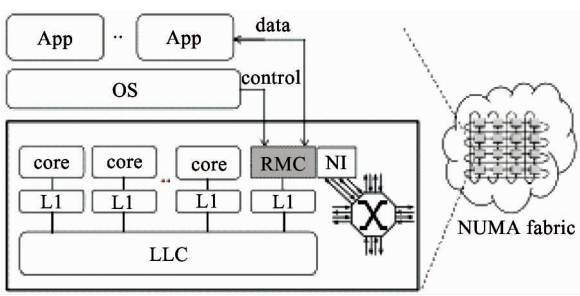

图 6　soNUMA 架构示意图[41]

2.3　INFINISWAP 内存共享架构

密歇根大学的 Gu 等人[42]提出了一种架构兼容的内存松耦合共享方式,并将该工作命名为 INFINISWAP。每台服务器节点上都有一个软件进程负责管理的空闲内存资源,这些内存资源被映射到硬盘交换空间的同时被划分为多个块。这些内存块可以被其他的服务器节点通过 RDMA 网络在不经过中央处理器(central processing unit,CPU)的情况下直接访问。数据在以同步的方式写到映射的内存块的同时也会以异步的方式写到服务器本地的硬盘空间以达到备份数据的目的,这样可以大大提高容

错率,一旦由于网络故障等原因造成远程数据无法访问时,INFINISWAP 的后台进程会从本地硬盘空间读取备份数据恢复。

INFINISWAP 有 2 个优势。其一是架构兼容性,无需任何新的计算机体系架构、新的硬件设计以及新的软件编程框架,只需要部署软件进程在支持 RDMA 的集群上就可以实现内存松耦合共享。另一个优势是不需要中心节点的调度,INFINISWAP 内部实现了一种内存空间的管控机制,由空闲内存使用节点和贡献节点直接协商。

作者在真实的服务器上使用大数据应用进行了评估,结果显示相比于使用磁盘空间,使用 INFINISWAP 借用远程内存可以实现 4 ~ 15 倍的性能提升。

2.4 Venice 松耦合数据中心架构

来自中国科学院计算所的侯锐研究团队从成本有效的角度设计了数据中心服务器的架构,相关工作发表在 2013 年的 HPCA 会议上[43,44]。该工作使用 PCI-e 协议和交换芯片构建了一个多节点系统,在该系统上可以实现内存、网卡和 GPU 的共享借用。相比于基于以太网的资源共享系统,该系统可以达到 5 倍的内存访问带宽和 1/12 的访问延迟。这篇文章有效地实现了资源再分配,达到降低数据中心成本的目的。

基于 PCI-e 的资源共享系统存在 2 个问题。第 1 点是由于 PCI-e 协议的访问延迟和协议转换的软件开销,在访问其他节点的远程资源时有不可忽略的性能损失。第 2 点是受限于多次的协议转换,需要复杂的软件配合才能实现不同远程资源的访问,其兼容性和可扩展性较差。

计算所侯锐研究团队基于以上工作在 2016 年的 HPCA 会议上提出了 Venice 松耦合数据中心架构,如图 7 所示[45]。Venice 架构从数据共享机制出发,提出了一套包含 3 个层次的数据中心架构。首先针对远程设备的访问延迟和带宽需求,定制了多模态的高速通信协议。其次,针对现有软件对通信协议的适应性问题,定义了资源访问机制作为软件和硬件资源的系统接口。最后,从数据中心整体出发,统一资源调度,优化资源管理策略。文献[45]介绍了基于 FPGA 的原型验证平台,实验结果显示通过 Venice 架构访问远程内存、远程网卡、远程加速器资源时都可以获得很好的性能提升。

Venice 架构是学术界第一个完整提出松耦合数据中心架构的文献,其贡献主要包括:(1)设计了完整的松耦合数据中心架构方案,实现了多种资源池(内存、网卡、加速器)的共享方案。(2)通过 FPGA 实验平台论证了松耦合数据中心架构设计中经常被忽略的访问延迟的重要性。(3)设计了完整的软硬件架构,可以完美移植当前数据中心应用无需软件修改。

Venice 架构受到华为计算所战略合作项目的资助,其核心研究成果(包括总体架构、资源共享协议等)被作为核心要素写入了华为 HTC-DC 的技术白皮书[19]。

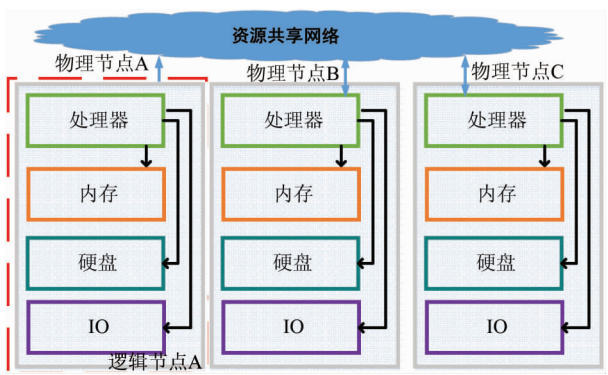

图 7 Venice 资源共享示意图

2.5 基于以太网和 InfiniBand 的节点间内存共享

国内外学者都做过远程内存访问的相关研究,他们将远程空闲内存用作交换空间、文件缓存和内存盘(ramdisk)等。这些研究工作都是基于传统的网络互连技术,如以太网和 InfiniBand 网络。

俄亥俄州立大学的研究者建议通过 InfiniBand 来使用远端内存[46]。他们在使用 InfiniBand 互连的集群中设计了一种能够利用远端内存的内存页系统。希腊伊拉克里翁研究和技术基金会计算科学研究所尝试使用低成本的商用网络和商用操作系统来获得更好的 I/O 性能[12]。他们仔细地评测了通过具有远程直接内存访问功能(RDMA)的商用网络来使用远程块设备的方法,发现整体性能受到中断开

销、请求大小和协议消息大小的限制。他们在研究工作中使用的是具有远程直接内存访问功能的万兆以太网。该研究所还设计、实现并评测了一个网络内存盘[13]。他们使用远端内存作为更快的硬盘存储。这种网络内存盘是可移植的,具有良好的灵活性并且能够在任何已有的 Unix 文件系统中使用。斯沃斯莫尔学院针对异构的 Linux 集群提出了一个网络交换系统,并命名为 Nswap[14]。Nswap 是一个基于 TCP/IP 协议的可卸载内核模块,具有良好的时间效率和空间效率。

华中科技大学提出了一个方法可以让虚拟机利用集群内的空闲内存[47],这种方法基于以太网连接,通过利用集群内其他节点的空闲内存来克服单台机器物理内存的限制。他们的方法可以有效地减少内存密集型和 IO 密集型应用的执行时间。湖南大学设计实现了一个分布式系统允许用户透明的访问远端内存[15]。

2.6 小结

表 1 从是否需要修改软硬件架构到支持的松耦合资源类型对当前工业界和学术界松耦合数据中心架构的特点进行了总结对比。可以看出,越多的远程资源访问类型支持需要越复杂的设计,相应需要修改的硬件架构和软件程序越多。

3 松耦合数据中心的特点

虽然工业界和学术界对于松耦合数据中心的研究存在一些不同,但是核心思想都是相同的,即通过资源的松耦合化提高数据中心资源利用率并降低成本。本节内容将提炼总结松耦合数据中心的主要特点。

表 1 松耦合数据中心架构比较

项目名称	是否修改硬件	是否需要修改用户程序	是否支持远程内存访问	是否支持远程加速器访问	是否支持远程网卡访问	发布年份
RSA	√	√	√	×	×	2013
Catapult	√	√	×	√	√	2014
Open Compute Project	√	√	√	√	×	2011
HTC-DC	√	√	√	√	√	2014
Gen-Z	√	√	√	√	√	2016
rCUDA	×	×	×	√	×	2012
DSM	√	√	√	×	×	1988
Disaggregated Memory	√	×	√	×	×	2009
soNUMA	√	√	√	×	×	2014
INFINISWAP	×	×	√	×	×	2017
Venice	√	部分需要	√	√	√	2016
基于以太网的内存共享	×	×	√	×	×	
基于 InfiniBand 的内存共享	×	×	√	×	×	

3.1 承载数据访问的通信机制

远程资源的访问是以通信机制传输的数据包为载体的,现有的主要通信机制集中在以太网和一些 I/O 通信协议,例如 InfiniBand、ZigBee 和 PCI-e 等通信协议。松耦合数据中心架构如图 8 所示。这些通信机制通过软硬件协同工作可以达成远程资源访问的目的,然而它们在性能表现上存在一定问题。通信机制的表现直接关乎到远程资源访问效率,所以也有研究人员针对松耦合数据中心架构设计了专用的通信协议[38]。

在设计松耦合数据中心通信机制时需要考虑以下 2 点:

(1)数据中心应用种类飞速增长,不同的应用对于资源需求的差异性也愈发明显,而且相同应用对资源的需求也存在时变性。即在程序运行的不同阶段,应用对于计算、内存、网络等资源的需求量存

在很大差异。这 2 个特点要求松耦合数据中心的通信机制具备动态调整能力,能够满足不同应用的访问需求,可以在延迟和带宽之间以及不同的访问粒度之间做到动态平衡。

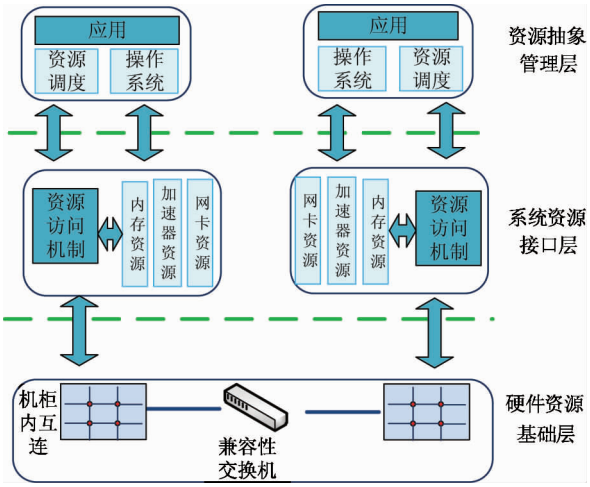

图 8　松耦合数据中心架构示意图

(2)松耦合数据中心的通信机制还需要考虑互连逻辑和网络连接之间的影响。考虑到访问部分远程资源时的延迟需求,互连逻辑期望相连节点间的距离越短越好。一方面,通信互连所提供的可用链路数据的增加可以构建更高阶的互联网络,有效降低传输距离,但是也相应地导致硬件成本的提高或者链路带宽的损失。另一方面,给定互连逻辑可以链路数量,连通度更好的网络可以降低节点间的最大距离,但网络连通度的增加则可能会限制网络的最大规模。理论上,全互连网络可以最小化节点间距离,但其所支持的网络规模也是最低的。合理的通信机制需要充分考虑互连逻辑和网络连接的关系,能够在延迟、带宽以及硬件成本之间进行有效的权衡。

3.2　接通软硬件的资源访问机制

通信机制是松耦合数据中心进行资源共享的硬件基础,此外还需要大量系统软件模块与之配合。这些系统软件模块承接物理硬件和用户软件的交互工作,是访问远程资源必不可少的一部分。通过之前的工作介绍,可以看到任何松耦合数据中心的架构设计都有相应的系统软件配合。松耦合数据中心中的远程资源访问需求由资源访问机制转换成相应的控制指令传递给硬件通信机制,资源访问机制还需要把来自硬件通信机制的数据和控制指令传递给相应的硬件资源和数据中心软件应用。

资源访问机制在设计的过程中也有 2 个方面需要考虑:

(1)要尽量利用现有操作系统和底层软件的接口和机制,避免或减少软件的修改,这是为了能够更好地兼容现有数据中心架构和硬件基础。

(2)要提供用户级应用访问的透明性,尽可能使用户无需感知远程资源的存在。数据中心应用种类繁多且更新较快,无法为了不同的数据中心架构做一一适配。良好用户级应用访问的透明性是松耦合数据中心兼容性和可扩展性的必要条件。

3.3　负责全局调度的资源管理机制

通信机制和资源访问机制,目的在于构建本地操作系统和远程资源之间的桥梁,支持点到点的资源访问,保证远程资源访问请求的可达性和远程资源的可用性。在这一过程中还有一个重要的角色就是全局的资源调度者,全局的信息收集和资源调度能够为不同的资源访问需求分配合理的空闲资源,实现整体效率的提升。负责全局调度的资源管理机制有 2 个任务,即资源信息收集和资源分配调度。

资源信息收集可以通过 2 种方式实现,即集中式设计和分布式设计。集中式的资源收集机制通过主从式的资源信息校核,由主控节点获取全局的资源信息。该机制具有设计复杂度较低、信息设计效率较高的优点,但也面临可靠性较低的问题,很有可能成为系统瓶颈。分布式的资源收集机制则通过与相邻节点交互获取全局资源信息。该机制的扩展性和可靠性都有很明显的优势,但其设计复杂度较高,并且信息传播的速度较慢,还有可能因为信息更新不及时导致资源分配效率的损失。

由于数据中心资源属性的差异性,比如位置、性能、容量、链路带宽利用率等,资源效率的发挥很大程度上决定于资源的分配和调度策略。显然,合理的资源分配调度策略可以最大化资源的效率,而不合理的资源分配策略必将导致系统性能的损失。为最大化各种资源的利用率,充分发挥各种资源的效用,需要对系统资源的分配和调度策略进行研究。

3.4　保障资源共享的备份机制

数据中心的备份机制可以有效防止系统故障甚

至人为失误造成的数据损失以及经济损失。现有的数据中心备份机制主要由完全备份、增量备份、差量备份、实时备份 4 种策略[48-54]。

松耦合数据中心架构的备份机制区别于传统的数据中心备份机制,它不仅需要考虑正常的计算和存储数据的备份,还需要考虑备份使用远程资源时产生的数据,而且这部分数据不仅存在于本地节点也会存在远程资源所在的节点。设计一套完备的松耦合数据中心备份机制需要考虑以下 2 点:

(1)要针对不同存储位置的数据制定不同的备份策略,在保障完备性的同时也要尽量减少空间占用。松耦合数据中心架构的特点造成需要备份的数据可以分布在各部分的硬件存储中,例如本地节点的内存和硬盘,远程加速器资源的存储部件,甚至包括硬件通信部件的缓存空间。合理利用这些数据存储位置和特点能够有效提升备份机制的效率。

(2)设定不同层级的备份和恢复策略。例如由于硬件通信错误导致的数据传输中断应该在硬件层面完成数据备份和恢复,操作系统甚至用户软件介入这一过程会带来不必要的性能开销。

4 结 论

本文详述了松耦合数据中心架构的研究背景,梳理了工业界的发展现状和学术界的研究成果,分析总结了松耦合数据中心架构的技术特点。松耦合数据中心架构是数据中心发展的重要趋势,已经获得了工业界和学术界的广泛认可。但松耦合数据中心架构的研究仍面临许多挑战,工业界和学术界的研究人员可以从以下 3 个方面展开研究:

(1)可以根据用户应用资源需求动态调整链路状态的通信机制。链路状态动态调整范围不仅包括延迟带宽等参数,还可以包括网络拓扑和链接数量等信息。

(2)完全兼容的资源访问方式,对用户应用完全透明,现有的数据中心应用无需修改就可以部署在松耦合数据中心架构上,而且还要享受到松耦合数据中心动态资源共享带来的性能提升。

(3)更加智能的资源分配和回收机制,能够时刻保障整个松耦合数据中心的使用效率和性能输出。可以结合高速发展的人工智能技术,通过大量的数据收集智能地完成全局资源调度。

参考文献

[1] IDC 圈. 2017 年中国 IDC 产业发展研究报告[EB/OL]: http://www.idcquan.com/Special/2017baogao/: 中国 IDC 圈, 2017

[2] Jiang T, Hou R, Zhang L X, et al. Micro-architectural characterization of desktop cloud workloads[C]//2012 IEEE International Symposium on Workload Characterization (IISWC), San Diego, USA, 2012: 131-140

[3] Malladi K T, Nothaft F A, Periyathambi K, et al. Towards energy-proportional datacenter memory with mobile DRAM[C]//2012 39th Annual International Symposium on Computer Architecture (ISCA), Portland, USA, 2012: 37-48

[4] Ferdman M, Adileh A, Kocberber O, et al. Clearing the clouds: a study of emerging scale-out workloads on modern hardware[C]//17th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'12), London, UK, 2012: 1-11

[5] Reddi V J, Lee B C, Chilimbi T M, et al. Web search using mobile cores: quantifying and mitigating the price of efficiency[C]//International Symposium on Computer Architecture, Saint-Malo, France, 2010: 314-325

[6] Tang L, Mars J, Vachharajani N, et al. The impact of memory subsystem resource sharing on datacenter applications[C]//International Symposium on Computer Architecture, San Jose, USA, 2011: 283-294

[7] Nishtala R, Fugal H, Grimm S, et al. Scaling memcache at facebook[C]//the 10th USENIX Symposium on Networked Systems Design and Implementation, Lombard, USA, 2013: 385-398

[8] Jiang T, Zhang Q, Hou R, et al. Understanding the behavior of in-memory computing workloads[C]//2014 IEEE International Symposium on Workload Characterization (IISWC), Raleigh, USA, 2014: 22-30

[9] Jiang T, Hou R, Dong J, et al. Adapting memory hierarchies for emerging datacenter interconnects[J]. *Journal of Computer Science and Technology*, 2015, 30(1): 97-109

[10] Ren Z J, Xu X H, Wan J, et al. Workload characterization on a production Hadoop cluster: a case study on Taobao[C]//2012 IEEE International Symposium on Workload Characterization (IISWC), San Diego, USA, 2012: 3-13

[11] Mellanox Technologies. Introduction to Infiniband[EB/OL]. http://www.mellanox.com/pdf/whitepapers/IB_Intro_WP_190.pdf: Mellanox, 2003

[12] Marazakis M, Xinidis K, Papaefstathiou V, et al. Efficient remote block-level I/O over an RDMA-capable NIC [C]//Proceedings of the 20th Annual International Conference on Supercomputing, New York, USA, 2006: 97-106

[13] Flouris M D, Markatos E P. The network RamDisk: using remote memory on heterogeneous NOWs[J]. *Cluster Computing*, 1999, 2(4): 281-293

[14] Newhall T, Finney S, Ganchev K, et al. Nswap: a network swapping module for Linux clusters[C]//Euro-Par 2003 Parallel Processing, Klagenfurt, Austria, 2003: 1160-1169

[15] Zhou L, Wu S, Shi X, et al. An approach to use cluster-wide free memory in virtual environment[C]//Proceedings of the 2011 6th Annual China Grid Conference, Washington DC, USA, 2011: 163-167

[16] Mohan J. Kumar. Rack scale architecture for cloud[EB/OL]. https://packetpushers. net/wp-content/uploads/2014/09/Rack-Scale-Architecture-% E2% 80% 93-Platform-and-Management-SF14 _ DATS008 _ 104f. pdf :Intel IDF, 2013

[17] Rousset S. Accelerating time to value with future-ready rack-scale infrastructure [EB/OL]. https://blog. dellemc. com/en-us/accelerating-time-to-value-with-future-ready-rack-scale-infrastructure/: DellEMC, 2016

[18] Putnam A, Caulfield A M, Chung E S, et al. A reconfigurable fabric for accelerating large-scale datacenter services[J]. *IEEE Micro*, 2015, 35(3): 10-22

[19] Fowers J, Kim J Y, Burger D, et al. A scalable high-bandwidth architecture for lossless compression on fpgas [C]//2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines, Vancouver, Canada,2015: 52-59

[20] Ovtcharov K, Ruwase O, Kim J Y, et al. Accelerating deep convolutional neural networks using specialized hardware[J]. *Microsoft Research Whitepaper*, 2015, 2(11): 1-4

[21] Caulfield A M, Chung E S, Putnam A, et al. A cloud-scale acceleration architecture[C]//The 49th Annual IEEE/ACM International Symposium on Microarchitecture, Taipei, China, 2016: 1-13

[22] Chung E, Fowers J, Ovtcharov K, et al. Serving DNNs in real time at datacenter scale with project brainwave [J]. *IEEE Micro*, 2018, 38(2): 8-20

[23] Fowers J, Ovtcharov K, Papamichael M, et al. A configurable cloud-scale DNN processor for real-time AI[C]// Proceedings of the 45th Annual International Symposium on Computer Architecture, Los Angeles, USA, 2018: 1-14

[24] Firestone D, Putnam A, Mundkur S, et al. Azure accelerated networking: SmartNICs in the public cloud[C]// 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), Renton, USA, 2018: 1-14

[25] Open Compute Project[EB/OL]. http://opencompute. org/: OCP, 2018

[26] HUAWEI. High throughput computing data center architecture[EB/OL]. http://www. huawei. com/ilink/en/download/HW _ 349607: Huawei, 2014

[27] HUAWEI. Huawei release smart DC 3.0 @ AI at CEBIT to bring AI into data centers[EB/OL]. https://www. huawei. com/en/press-events/news/ 2018/6/ huawei-smart-data-center-3-0-ai-at-cebit. :Huawei, 2018

[28] Gen-Z. Consortium[EB/OL]. http://genzconsortium. org/: Gen-Z, 2018

[29] rCUDA[EB/OL]. http://www. rcuda. net/ : rCUDA, 2018

[30] Li K. IVY: a shared virtual memory system for parallel computing[C]//Proceedings of the International Conference on Parallel Processing (ICPP88), The Pennsylvania State University, University Park, USA, 1988: 94-101

[31] Amza C, Cox A L, Dwarkadas S, et al. TreadMarks: shared memory computing on networks of workstations [J]. *IEEE Computer*, 1996, 29(2):18-28

[32] Scales D J, Gharachorloo K, Thekkath C A. Shasta: a low overhead, software-only approach for supporting fine-grain shared memory[J]. *ACM SIGOPS Operating Systems Review*, 1996, 30(5): 174-185

[33] Bershad B N, Zekauskas M J, Sawdon W A. The midway distributed shared memory system[C]//Proceedings of the 38th IEEE International Computer Conference, San Francisco, USA, 1993: 528-537

[34] Keleher P J. The relative importance of concurrent writers and weak consistency models[C]//Proceedings of the 16th International Conference on Distributed Computing Systems, Hong Kong, China, 1996: 91-98

[35] Hu W, Shi W, Tang Z, et al. JIAJIA: a software DSM system based on a new cache coherence protocol[C]// International Conference on High-Performance Computing and Networking, Berlin, Germany, 1999: 461-472

[36] Agarwal A, Bianchini R, Chaiken D, et al. The MIT Alewife machine: architecture and performance[C]//The 22nd Annual International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, 1995: 2-13

[37] Laudon J P, Lenoski D E. The SGI origin: a ccNUMA highly scalable server[C]//The 24th Annual International Symposium on Computer Architecture, Denver, USA, 1997, DOI: 10.1145/264107.264206

[38] The versatile SMP (vSMP) architecture and solutions based on vSMP foundation. [EB/OL] http://www. scale-mp. com/prod/technology/how- does-it-work/: ScaleMP, 2016

[39] Lenoski D E, Laudon J P, Joe T, et al. The DASH prototype: logic overhead and performance[J]. *IEEE Transactions on Parallel and Distributed Systems*, 1993, 4(1): 41-61

[40] Lim K T, Chang J, Mudge T N, et al. Disaggregated memory for expansion and sharing in blade servers[J]. *ACM SIGARCH Computer Architecture News*, 2009, 37(3): 267-278

[41] Novakovic S, Daglis A, Bugnion E, et al. Scale-out NUMA[J]. *Architectural Support for Programming Languages and Operating Systems*, 2014, 49(4): 3-18

[42] Gu J, Lee Y, Zhang Y, et al. Efficient memory disaggregation with infiniswap[C]//The 14th USENIX Symposium on Networked Systems Design and Implementation NSDI 17, Boston, USA, 2017: 649-667

[43] Jiang T, Hou R, Zhang L H, et al. Using remote memory in data center with PCIe fabric[C]//Efficient Data Center Server workshop, Shenzhen, China, 2013

[44] Hou R, Jiang T, Zhang L H, et al. Cost effective data center servers[C]//Proceedings of the 19th IEEE International Symposium on High Performance Computer Architecture (HPCA), Shenzhen, China, 2013: 179-187

[45] Dong J B, Hou R, Huang M, et al. Venice: exploring server architectures for effective resource sharing[C]//High Performance Computer Architecture (HPCA), Barcelona, Spain: 2016: 507-518

[46] Liang S, Noronha R, Panda D K, et al. Swapping to remote memory over InfiniBand: an approach using a high performance network block device[C]//2005 IEEE International conference on cluster computing, Boston, USA, 2005: 1-10

[47] Deng Y, Sun J, Chen H, et al. Design and implementation of a distributed system for transparent remote memory accessing[C]//International Conference on Consumer Electronics, Yichang, China, 2012: 3163-3166

[48] Lu P, Zhang L, Liu X, et al. Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks[J]. *IEEE Network*, 2015, 29(5): 36-42

[49] Li D, Guo C, Wu H, et al. FiConn: using backup port for server interconnection in data centers[C]//International Conference on Computer Communications, San Francisco, USA, 2009: 2276-2285

[50] 张艳,李舟军,何德全. 灾难备份和恢复技术的现状与发展[J]. 计算机工程与科学, 2005(2):107-110

[51] 韩德志,谢长生,李怀阳. 存储备份技术探析[J]. 计算机应用研究, 2004(6):1-4 +7

[52] 顾鹏,刘立刚,谢长生. 数据存储系统备份技术研究与分析[J]. 计算机安全, 2003(6):71-72

[53] 戴士剑,张杰,郭久武. 数据恢复技术综述(上)[J]. 信息网络安全, 2006(1):47-49

[54] 徐伟,朱旭东,刘浏. 基于备份的 RAID5 在线重构框架[J]. 高技术通讯, 2012,22(1):28-34

A survey of disaggregated data center architecture

Zhao Boyan[*,**,***], Hou Rui[*], Zhang Qianlong[**,***], Bao Yungang[**], Zhang Lixin[**], Meng Dan[*]

([*] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093)

([**] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([***] University of Chinese Academy of Sciences, Beijing 100049)

Abstract

As the foundation of the information industry, the data center has experienced rapid growth in recent years and faces new challenges brought by big data. The mismatch between the traditional data center architecture and the emerging application features of the data center is becoming more apparent. At the same time, the resource utilization of the data center continues to be low, but the cost and power consumption of the data center are constantly rising. Faced with these problems, the development of disaggregated data center architecture has become an important development trend of data center. Disaggregated data center resources can break through traditional physical boundaries, improve resource utilization, and strengthen cost control. This paper summarizes the development status of disaggregated data center architecture, and refines the key technologies of the disaggregated data center architecture, and looks forward to its future development prospects.

Key words: data center, disaggregation, communication mechanism, resource sharing, resource allocation, data backup, computer architecture, field programmable gate array (FPGA)