一种现场、弹性、自治的网络服务系统 —— 海服务系统研究与设计

王劲林^{①*}, 田静^②, 尤佳莉^①, 刘学^①, 邓浩江^①

① 国家网络新媒体工程技术研究中心, 中国科学院声学研究所, 北京 100190

② 中国科学院信息工程研究所, 北京 100093

* 通信作者. E-mail: wangjl@dsp.ac.cn

收稿日期: 2014-10-30; 接受日期: 2015-04-09; 网络出版日期: 2015-09-15

中国科学院战略性先导科技专项“海云创新试验环境构建与关键技术研究”(批准号: XDA06010000)、“未来网络架构研究与边缘设备研制”(批准号: XDA06010302)、“智能电视平台与服务支撑环境研制”(批准号: XDA06040501)和中国科学院声学研究所所长择优基金“海计算——一种弹性、可自治的现场计算方法研究和应用”(批准号: Y454231601)资助项目

摘要 本文针对网络发展以用户为中心的趋势, 研究以融合网络边缘设备为基础, 面向具有实时性等高 QoS 要求的网络服务, 在广域环境下, 构建弹性自治现场处理系统——海服务系统, 为每个服务请求动态、弹性地提供面向其特征和 QoS 需求的海服务, 从而满足多类型网络服务高效处理的要求. 另外, 融合海服务、云计算各自特点, 设计适用于广域分布式环境和大数据分级处理的海云协同计算网络架构, 形成一个通用的以海服务为中心的海云协同服务平台, 为实时处理服务, 特别是大数据处理和分析, 提供数据高效汇聚、模型实时计算和 QoS 多级服务等性能保障, 为网络服务提供更高效率的处理机制.

关键词 海服务 实时服务 服务调度 海云协同 数据处理

1 引言

随着信息技术的飞速发展, 尤其是互联网、移动互联网、物联网的普及和应用, 人类社会(人)与信息世界(机)、物理世界(物)的三元融合趋势凸显, 在给传统行业和新兴技术带来新的机会、平台和途径的同时, 也给传统信息技术体系带来了“能耗、安全、传输、处理”4个方面的挑战, 应对这些挑战, 特别是当信息的获取量大到足以使传输成为瓶颈时, 很多信息可以在前端处理和存储^[1], 中国科学院在战略性科技先导专项“面向感知中国的新一代信息技术研究”中, 提出了利用分布在客户端的计算能力来提供具有个性化、现场化特点的海服务的概念, 以海云协同的模式构建新一代的信息服务体系.

应对复杂任务或服务的快速处理, 目前主要采取两种方式, 一是并行计算, 伴随并行机的发展而兴起, 其主要目标在于, 在并行机上解决一批具有重大挑战性计算任务的科学、工程及商业计算问题, 满足不断增长的应用问题对速度和内存资源的需求. 并行计算比较显著的特点是关注并行机上的核间调度, 优化任务完成时间, 并不考虑广域网络环境下, 具有网络连接特性的处理器间协同处理问题; 另一

引用格式: 王劲林, 田静, 尤佳莉, 等. 一种现场、弹性、自治的网络服务系统 —— 海服务系统研究与设计. 中国科学: 信息科学, 2015, 45: 1237–1248, doi: 10.1360/N112014-00344

个则是云计算, 其主要通过虚拟化技术, 将硬件资源, 如 CPU、内存、存储, 构建形成弹性资源池, 按需提供给用户. 有研究者称, 云计算可看作除水、电、气和电话外的第 5 类资源^[2]. 目前, 云计算从集中式数据中心, 到分布式云, 从数据采集存储, 到大数据分析处理, 云计算平台的形态和规模不断演进, 复杂性也显著增加. 而面对发展迅猛的移动互联网服务和物联网应用, 由用户和端设备产生的大量数据需要实时的采集、传输和处理, 给当前的网络系统以及各种计算系统带来了巨大压力, 主要体现在:

(1) “云” 远在天边. 云计算取代了能力有限的单节点服务器, 使用虚拟化的方式对资源进行整合, 由于云计算中的服务器从网络上看大多远离用户, 对于实时性需求较高的服务, 如物联网、安全防护、媒体服务等, 需要在用户服务请求发起后, 服务端能够即时响应, 为其提供现场 (on-site) 的即时服务, 从而保障服务的高效处理和即时响应. 因此, 云计算的目标和实现方法难以满足这些类型服务的需求.

(2) 大数据难以高效汇聚和快速处理. 云计算技术适合需要全局监督和优化调度、易于批处理的应用, 并行计算应用于计算密集型, 特别是科学计算等领域. 当大规模的用户或端设备实时产生的海量数据无法快速上传到云端、并行计算机或集群时, 就难以实施实时的分析和处理、无法高效处理具有实时性或者短周期模型更新等需求的大数据分析相关应用.

因此, 如何通过现场计算快速处理局部区域的实时信息, 并与云端协同优化服务质量, 是提高网络所承载应用的性能的有效方法.

综上分析, 在构建一个以用户为中心, 满足在线实时、准实时要求的应用时, 需要充分考虑网络和服务特性, 利用网络边缘设备, 提供现场 (on-site)、弹性 (elastic)、自治的 (autonomous) 服务, 我们把这种服务简称为海 (SEA) 服务. 其中, 现场指具有实时、准实时的服务请求, 可以得到快速响应; 弹性则针对计算、存储和网络等资源, 可根据服务需求动态调配; 而自治可满足大规模服务网络的自组织、管理以及支持网络的动态维护. 因此, 秉承海云协同服务的创新理念, 提出了一种面向即时响应需求的现场弹性自治服务网络结构 —— 海服务网, 其主要思想是: 通过汇聚网络边缘设备, 利用节点的自治管理和协作, 构建节点范围弹性可调、服务资源区域动态自治、服务交付点靠近服务请求发起端的服务网络. 从而以服务请求节点为中心, 实时构建面向该服务的处理网络, 提供可靠、实时的服务, 从而动态地为具有实时或者即时处理要求的服务请求提供高效的服务响应.

以融合网络边缘设备为基础, 面向网络多类型服务, 在广域环境下, 研究海服务网络建模、调度、管理等关键技术, 从而为具有 QoS 约束的网络请求按需提供服务, 满足不同网络服务即时响应、高效处理的要求, 降低服务响应时间, 提升服务处理效率, 降低网络服务拒绝率. 另外, 融合海服务、云计算各自特点, 研究和设计适用于广域分布式环境和大数据分级处理的海云协同计算网络架构, 形成一个通用的以海服务为中心的海云协同服务模型, 为实时处理服务, 特别是大数据处理和分析, 提供数据高效汇聚、模型实时计算和 QoS 多级服务等性能保障, 为网络服务提供更高效的处理机制.

本文研究内容为网络即时服务提供了新的处理方法, 无论对未来网络研究的推进、互联网应用性能的提高, 还是机器学习在互联网中的应用和在未来智能化生活的促进方面, 均有极大研究价值.

2 海服务网络关键技术分析

为了应对网络发展所带来的新的服务方式转变, 许多企业和机构均进行了相关部署和探索.

IBM 提出了业务前向转型 (front office transformation)¹⁾ 的概念, 其思想是着重考虑用户体验, 对服务模式进行转型. 思科提出了面向边缘服务的雾计算^{[3]2)}, 将服务部署在边缘, 以提升实时处理效

1) <http://www-935.ibm.com/services/us/business-consulting/transform/>.

2) <http://www.cisco.com/web/solutions/trends/tech-radar/fog-computing.html>.

率,提高服务性能.海服务是我们在以用户为中心的服务模式下,一个新的探索.中国科学院于2010年首次提出“海计算”的概念,在战略性科技先导专项“面向感知中国的新一代信息技术研究”中,对海云计算、海云存储和利用海终端的弹性处理等方面均开展了研究.文献[4]面向ZB级数据处理问题,提出了海云计算系统架构,其相关文章研究和探讨了包括一种基于REST结构扩展的新的计算模型^[5,6],管理ZB级数据的存储系统结构^[7,8],具有能量优化的数据中心服务^[9],以及弹性处理^[10,11]等多个问题.文献[12]面向海云结构下大数据处理中的关键问题展开研究,以区域性智能数据中心和高速互联网为基础设施,研究大规模数据的存储、处理、挖掘及可视化等技术,从而提供多种数据服务.

在上述已有的海云系统技术研究中,研究人员从系统、数据存储、处理器调度等多个角度开展了工作,并结合物联网传感器等网络边缘设备的特点,进行特定领域的研究.本文所提出的海服务网络,更多侧重于互联网中的边缘设备和资源,面向互联网中即时处理需求,从而可为不同服务场景,如视频服务、网络在线服务、安全防护等,提供一个通用的服务模型和架构,并对关键技术开展研究.因此,在海服务网络研究过程中,有许多新的问题需要进一步考虑,包括如何对网络进行自治管理及弹性提供海服务,如何在广域环境下对复杂服务进行高效调度和优化均衡,如何进行海服务和云计算的协同从而满足不同应用和大数据处理的需求等,这些均是海服务网络的技术难点.

(1) 海服务网络组网技术. 该技术的本质是广域环境下海端节点的自治管理和维护,与P2P网络和内容分发网络(CDN)^[13]中拓扑组织方法类似.在这些系统中,如何设计网络结构便于节点间内容共享是要解决的问题,因此大多数方法需要建立节点之间的覆盖网结构(overlay),利用覆盖网结构特点来完成资源的共享.在海服务网络组网中,在建立overlay的同时,需考虑其他额外因素,例如地域特征、延时特征等等,为不同的服务要求作准备.文献[14~23]根据不同网络特征,提出了overlay的节点组织方式,主要根据节点位置等信息构建节点组、自治域,或者建立多级聚合等方式,提高路由效率和降低维护成本.然而,大多数面向协作的网络拓扑结构、节点聚合分组算法以及基于内容的路由查找算法,仅考虑单维影响因素,如地理位置、网络距离、内容名字等.而对于海服务而言,为了提供高效服务,需要综合考虑网络状态、服务能力、服务请求QoS等多种特征,实时的对服务类型相似、延迟低、或者行为趋同的节点进行维护.而这个问题在以前的覆盖网络研究中并没有得到充分的关注,是海服务研究的一个重要方面.

(2) 广域环境下海服务调度技术. 现实中,一个完整的工作流可能由多个任务以一定逻辑依赖关系构成,考虑广域环境下服务节点间连接的实时状态和任务的依赖关系,如何将任务进行调度和处理,是一个重要的问题.

关于海服务节点调度,可以借鉴多核处理器的调度方法.目前已经有很多算法可以解决多核处理器上的任务调度问题^[24~32].其中,EDF算法和Pfair算法被广泛用于多核系统中的实时任务调度,但是,它们以及相关改进算法都难以直接应用于海服务调度中.例如,EDF算法基于抢占式任务调度环境,但是在实际应用中任务不一定是可抢占的,当系统中任务数量很多时频繁的任务抢占、切换会造成很大的系统开销;Pfair算法虽然可以应用于非抢占式任务调度中,但是一般来说Pfair算法要求调度的子任务有相同的执行时间,而在实际系统中,服务中可切分的子任务难以统一长度时,就需要将所有的子任务扩展为相同的长度才能使用该算法,这会导致明显的资源浪费和效率降低.除此之外,以上算法中均没有考虑广域环境下不同海服务节点间子任务的传输代价,难以直接用于海服务调度,而这部分是影响高效处理性能的关键.

(3) 海云协同处理技术. 文中所提的海服务网络,主要面向网络业务需求,提供高效、现场服务和大数据的局部处理,而云计算适用于全局数据汇聚和数据批量分析,因此,两种服务系统各有侧重.如

图 1 服务网络功能架构

Figure 1 The infrastructure of the sea service network

果与云计算平台进行协同应用, 可以将局部和全局信息进行分级处理, 为用户提供多类型、多级别服务, 满足请求的不同 QoS 要求. 而云服务如何协同网络边缘设备构成的海服务网络提供弹性、多类型服务, 是一个新的问题. 因此, 在海服务的研究中, 我们的目标是设计一个海云协同架构, 满足资源统一调度, 支持高效大数据分析, 提升用户体验.

3 现场、弹性、自治现场服务 —— 海服务网络架构及建模

网络中的边缘设备, 称为海端设备, 可能包括部署的服务设备、用户终端、其他边缘设备, 这些具有处理能力、且可以在空闲时为网络提供服务的节点的全集构成了海服务节点空间. 由于海服务节点的服务处理能力和节点间的数据交换能力差异较大且动态性强, 因此, 如何对海服务节点进行组网、根据现场环境为每个服务请求迅速生成一个候选节点集合, 并实现服务交付节点的优化调度和任务指派是海服务系统的关键.

为此, 设计了模块化的海服务系统功能架构, 在该结构下, 海服务空间节点可基于自身能力承载相应的功能, 以实现节点间的通信、组网以及资源管理与服务调度, 有机形成一个现场弹性自治的网络服务系统 —— 海服务系统, 从而满足互联网中不同类型服务, 以及服务资源弹性调度的要求.

3.1 海服务系统功能架构

海服务系统是一个面向网络服务的通用平台, 通过该系统, 可以将所有提供海服务的节点进行自治管理, 并进行资源的监测和网络状态测量, 按需的功能加载和资源调配, 同时支持海服务全生命周期的管理. 该系统主要由 workflow 处理、服务调度管理、候选节点集合管理、任务执行环境、节点资源管理、连接资源管理、服务寻址、海空间管理、网络与设备管理以及安全等功能组成, 其结构见图 1.

海空间节点可根据需要承载不同功能, 至少包括任务执行环境、节点资源管理、连接资源管理功能及服务寻址、海空间管理、网络与设备管理的相关功能, 只包括上述功能的节点称之为服务执行节

点, 还具有调度功能的节点称为服务调度节点, 各层的功能描述如下:

工作流处理 (workflow processing) 是海服务对外接口, 接收外部请求, 将其分解为由一系列有逻辑依赖关系的任务 (task), 并按逻辑依赖关系, 输出到服务调度管理层, 并将外部请求的最终结果返回请求端.

服务调度管理 (service scheduling management) 负责任务的调度, 在为每个 task 形成了一个候选的协作节点集合后, 负责选择合适的任务执行节点, 并将任务加载到所选择的节点, 其中的服务调度算法是整个海服务系统能否保障服务性能的关键问题之一, 需要考虑服务优先级、候选服务节点状态、网络实时状态、QoS 要求等多方面因素. 其中, 候选节点集合管理是根据 task 的类型, 从节点的邻居表中选出了一个子集, 形成一个可执行该 task 资源、与请求节点间的数据交换代价足够小的候选节点集合.

任务执行环境 (task runtime) 接收服务调度管理下发的任务, 执行任务并监视任务执行状态, 通过输出定向把任务执行结果输出到下一跳. 任务执行环境通过与节点资源管理层和连接资源维护层间的接口感知节点资源的能力, 并能合理分配和管理资源, 包括新服务类型的注册、生成、服务加载、服务类型与底层资源的映射等功能. 这里的资源主要包含节点本身的资源及连接资源.

节点资源管理 (node resource management) 主要负责对节点资源、节点服务支持特性、节点资源占用和任务 — 资源映射等信息进行管理和维护.

连接资源管理 (connection resource maintaining) 负责网络信息探测、网络连接特性分析和预测、节点信息交换、overlay 维护等工作, 测量信息包括: 双向链路带宽、延迟、可用性、可支持服务类型以及其他网络连接相关信息等.

海空间管理 (SEA space management) 主要负责对网络中可参与服务的所有海服务节点进行管理, 包括节点加入/退出机制.

服务寻址 (service locating) 为外部请求提供全局负载均衡或重定向.

此外, 网络与设备管理 (network and equipment management) 即通常说的网管系统; 安全 (security) 功能则负责安全防护、攻击检测等.

在整个海服务网络系统中, 工作流处理是整个海服务的对外接口, 外部请求分解后的每个 task 处理, 是海服务建模和海服务调度所要解决的问题. 3.2 和 3.3 小节分别针对以 task 为中心的海服务处理进行详细探讨.

3.2 海服务网络建模

海服务网络建模是根据网络节点的属性特点, 对海服务的服务过程进行建模和分析, 该模型是满足网络服务的现场响应和弹性服务的核心.

在海服务网络系统中, 各节点自行维护供海服务选择的邻居节点集合, 当有任务请求到来时, 由具备调度功能的节点选择合适的任务执行节点, 通过输入定向接受任务执行所需数据以及控制信息, 在完成任务后, 通过输出定向将结果输出, 供后续任务使用.

上述海服务过程可通过下述模型表示:

假设海服务节点空间包含 N 个节点, 节点集合 $P = \{p_1, \dots, p_i, \dots, p_N\}$, 对应网络节点的属性集合 $\Sigma = \{\zeta_1, \zeta_2, \dots, \zeta_N\}$. 每个 $\zeta_i (1 \leq i \leq N)$ 实时维护节点 p_i 的相关信息, $\zeta_i = \{w_i, s_{up}, s_{cur}, c_{up}, c_{cur}, v_1, \dots, v_k, \dots, v_M\}$, 其中, w_i 表示节点 p_i 的可用带宽, s_{up} 和 s_{cur} 分别表示节点最大存储容量和当前已占用存储情况, c_{up} 和 c_{cur} 分别表示节点最大计算能力和当前已占用的计算能力, v_k 表示节点 p_i 处理

服务类型 k 对应的服务能力度量, M 为海服务节点集合中所有节点能完成的所有服务类型总量. $\tau_{i,k}$ 表示节点 p_i 是否具备当前请求所需服务类型 k 的服务能力, 其中

$$\tau_{i,k} = \begin{cases} 1, & \text{节点 } p_i \text{ 可执行服务类型 } k; \\ 0, & \text{节点 } p_i \text{ 不可执行服务类型 } k. \end{cases}$$

1) 网络中的某一节点 p_i 接收到一个服务请求 r , 其信息包括, 对应服务类型 k_r , 对应的处理需求 σ_{k_r} , 数据量 ψ_{k_r} , 服务响应时限 t_{k_r} .

2) 针对请求 r 的任务特征和 QoS 要求, 接收到请求的节点 p_i 通过其维护的网络中的有序邻域节点集合和邻域信息表, 通过候选处理节点选择函数进行选择, 形成一个规模为 $N_{i,r}$ 的候选节点集合 $\sum_i^r$, 其中, 节点集合规模 N_{k_r} 可以通过预设的参数或者自学习的方式得到, 邻域节点集合是根据邻域节点集合中的点到节点 p_i 可用带宽从大到小排序的有序集.

节点选择函数描述如下:

$$\begin{aligned} \sum_i^r &= F\left(\sum\right) = F_3\left(F_2\left(F_1\left(\sum\right)\right)\right), \\ F_1\left(\sum_i^r\right) &= \sum_{i,1}^r = \left\{ \zeta_j \in \sum_i^r \mid t_{i,j,k_r} = T(i,j,\sigma_{k_r}) + D(i,j,\psi_{k_r}) \leq \min(T(i,j,\sigma_{k_r}), t_{k_r}) \right\}, \\ F_2\left(\sum_{i,1}^r\right) &= \sum_{i,2}^r = \left\{ \zeta_j \in \sum_{i,1}^r \mid \sum_{i,1}^r \text{ 中按 } t_{i,j,k_r} \text{ 从小到大排序取前 } N_{k_r} \text{ 个} \right\}, \\ F_3\left(\sum_{i,2}^r\right) &= \sum_i^r = \left\{ \zeta_j \in \sum_{i,2}^r \mid \tau_{j,k_r} = 1 \mid p_j \text{ 可加载服务类型 } k_r \right\}, \end{aligned}$$

其中, 函数 $T(i,j,\sigma_{k_r})$ 表示节点 p_i 接收到的处理量为 σ_{k_r} 的任务 r 后, 如果由 p_j 进行处理的对应完成时间, 包括排队等待时间. $D(i,j,\psi_{k_r})$ 表示任务迁移到 p_i 上执行的总时间, 包括服务迁移时间及返回服务结果到输出目的地的数据传输时间, 这些值与网络双向带宽、处理前和处理后的数据总量、服务提供点、服务结果输出目的地相关. 如果这个时间足够小, 则将节点加入候选节点集合.

3) 根据任务时限, 进行节点调度, 这里节点调度方法通过计算选择函数的形式实现 (详情见 3.3 小节).

4) 如果步骤 3) 中选择的服务节点是当前服务调度节点, 则直接服务; 如果是候选服务节点列表中的点, 则进行任务迁移并处理.

5) 如果步骤 4) 中调度过程和任务处理过程均已完成, 则释放服务处理所占用资源, 更新节点当前状态信息, 并释放针对该服务的候选服务节点表.

在上述过程中, 候选节点选择采用计算选择函数的方式实现. 选择函数是指针对服务请求, 计算网络中其他节点执行该请求的 QoS 与服务接收节点执行该请求的 QoS 之间的差异, 其差异越小, 可有其他节点代替服务接收节点执行该请求的可能性越大. 在下一节, 我们针对调度特点, 提出了一个海服务调度方法.

3.3 海服务调度方法

在海服务网络中, 服务调度节点需要为所接收到的任务请求在候选节点集合中选择合适的执行节点, 从而满足资源的弹性调度和服务的 QoS 要求. 选择的原则是在满足资源利用率增量尽量小, 不同类型资源的占用尽量均衡的情况下, 根据多目标优化方法选择合适的节点. 其调度过程如下:

步骤 1 假设接收到服务请求的服务调度节点及其所有候选节点共同构成节点组 $P = \{p_1, \dots, p_j, \dots, p_M\}$, 节点组中的任意一个节点 p_j 的可用资源总量表示为 $p_j = \{s_{1,j}, s_{2,j}, \dots, s_{n,j}, \dots, s_{N,j}\}$, 节点 p_j 的已占用资源表示为 $po_j = \{so_{1,j}, so_{2,j}, \dots, so_{N,j}\}$. 其中, N 为一个节点组中所有节点一共可以支持的资源类型总数, $s_{n,j}$ 为 p_j 保持正常运行时位于该节点上的第 n 类资源的上限, 资源类型包括但不限于: cpu、内存、硬盘; 资源的上限值可以是系统上限, 也可以是预置上限. 服务调度节点接收到的服务请求 r 对应的任务资源需求向量表示为 $O = \{o_1, o_2, \dots, o_N\}$, 其中 o_i 表示服务请求 r 对节点组中第 i 类资源的总需求量; 任务所需资源, 可以通过系统预设值, 也可以通过历史信息进行计算和估计.

步骤 2 通过节点 p_j 在 N 种不同资源类型上的可用资源总量和占用量, 可知节点 p_j 的 N 种资源的剩余资源量, 并计算得到当前资源占用率向量 $rc_j = \{c_{1,j}, c_{2,j}, \dots, c_{N,j}\}$, 其中, 第 i 种资源的占用率为 $c_{i,j} = so_{i,j}/s_{i,j}$. 如果节点 p_j 对 r 进行处理, 则资源的占用率为 $ro_j = \{rp_{1,j}, rp_{2,j}, \dots, rp_{N,j}\}$, 其中, $rp_{i,j} = (so_{i,j} + o_i)/s_{i,j}$. 定义 $\delta_{i,j} = ro_{i,j} - rc_{i,j}$, 表示服务请求 r 加载到节点 p_j 后与加载之前比较第 i 种资源的占用率的增加量, 因此 r 对节点 p_j 的资源占用率增量向量为 $\Delta = \{\delta_{1,j}, \delta_{2,j}, \dots, \delta_{N,j}\}$, $\mu = \frac{\sum_{i=1}^N \delta_{i,j}}{N}$; 节点内资源均衡度指不同类型的资源占用率之间的方差, $\sigma_j = \frac{\sum_{i=1}^N (ro_{i,j} - \overline{rt_j})^2}{N}$, 其中, $\overline{rt_j} = \frac{\sum_{i=1}^N ro_{i,j}}{N}$.

步骤 3 如果节点 p_j 对 r 进行处理, 其第 i 种资源的占用率 $rp_{i,j} > TR_i$, 则将节点 p_j 从节点组中去除, 剩余节点组 $P' = \{p'_1, \dots, p'_j, \dots, p'_M\}$; 计算 r 对节点 p'_j 的资源占用率增量向量为 Δ 和 μ_j , 以及节点内资源均衡度 σ_j .

步骤 4 我们的目标是选出均衡度较好, 即 σ_j 较小, 同时资源占用影响较小, 即 μ_j 较小的点作为执行节点, 因此优化目标为选出合适的节点集合 P_k , 使其满足

$$P_k = \{p_k\} = \arg \min(\mu_j, \sigma_j)^T, j \in [1, M'].$$

上述问题是一个多目标优化问题, 可通过多种方法计算. 在这里, 通过如下步骤计算:

1) 定义均衡度搜索步长 $\Delta\sigma = (\sigma_{\max} - \sigma_{\min})/M'$, 其中 $\sigma_{\max}$ 和 $\sigma_{\min}$ 分别是节点组 P' 中所有节点对应 σ 值的最大值和最小值; 定义资源占用率增量步长 $\Delta\mu = (\mu_{\max} - \mu_{\min})/M'$, 其中 $\mu_{\max}$ 和 $\mu_{\min}$ 分别是节点组 P' 中所有节点对应 μ 值的最大值和最小值.

2) 以 $x \in [\mu_{\min}, \mu_{\max}]$, $y \in [\sigma_{\min}, \sigma_{\max}]$ 为搜索空间, 进行最优值搜索, 其中 $y(0) = \sigma_{\min}$, 如果没有节点满足 $\mu \ll x(t)$ 且 $\rho \ll y(t)$, 则 $x(t+1) = x(t) + \Delta\mu$, $y(t+1) = y(t) + \Delta\mu$, $t = t+1$, 循环上述过程; 如果有满足条件的节点 p_k , 则将该节点加入符合条件的节点集合 P_k , 形成调度节点, 终止上述过程, 输出节点集合信息.

通过上述过程得到一个符合条件的节点集合 P_k , 从中选择一个节点 p_k 作为最终决策点, 其选择方法可以通过多种选择策略, 例如: 随机、以某种概率分布或者以某种准则进行选择.

步骤 5 经过步骤 4, 所选出的节点 p_k 即是服务请求 r 的最终执行节点, 并由服务调度节点将 r 相关信息转发给节点 p_k 进行处理.

图 2 (网络版彩图) 海服务系统部署结构

Figure 2 (Color online) The deployment structure of the sea service system

4 海服务网络系统部署和典型应用分析

4.1 海服务系统部署架构

典型的海服务系统部署包括: 网管子系统、海空间管理子系统、服务寻址子系统、服务调度节点及服务执行节点, 服务调度节点和服务执行节点统称为海空间节点, 其中:

- 海空间管理子系统. 实现海空间节点结构化管理、服务的静态管理, 具体包括, 节点加入/退出机制及节点动态视图呈现; 服务与节点 Mapping 关系, 服务视图呈现.
- 服务寻址子系统. 为外部请求提供海空间中全局负载均衡或重定向.
- 网管子系统. 具备典型的网管功能, 配置、性能、告警及拓扑.
- 服务执行节点. 具有任务执行环境, 实现节点资源管理、连接资源管理功能, 以及具备海空间管理、服务寻址功能客户端模块.
- 服务调度节点. 除了服务执行节点功能及模块外, 还具备现 workflow 处理、服务调度管理、候选节点集合管理功能.

海服务系统典型部署结构如图 2 所示.

4.2 海云协同方案设计

海服务其基本思想是将服务请求通过靠近请求端的设备进行快速处理, 适合于局部快速计算和响应, 而云计算则是基于虚拟化技术, 汇聚处理资源, 对服务请求实现时空弱约束的承载, 两类计算模式各有侧重. 因此, 结合海服务实时性、局部性, 及云服务智能化的优势, 我们设计了一个海云协同服务系统, 为具有多等级用户体验和服务质量要求的应用提供解决方案. 其结构如图 3 所示, 通过构建海云协同管理平台对海端资源和云端资源进行统一调度和优化管理. 该平台可通过 workflow 引擎, 将应用解构为多个任务, 并优先将具有现场处理响应要求的任务指派到海服务系统, 通过海服务增强云计算的现场实时处理能力, 使网络服务既有云计算的服务资源自动化指派能力, 又有现场即时处理响应的功能, 应用仅需通过平台的开放接口请求服务, 即可得到包括现场即时响应在内的多等级用户体验和服务质量保障的服务. 该方案以网络服务能力前置部署和服务资源虚拟化融合的方式, 满足不同应用

图 3 (网络版彩图) 海云协同系统架构

Figure 3 (Color online) The structure of SEA-Cloud collaboration system

的需求, 提升网络服务的用户体验.

4.3 海云协同典型应用分析与展望

一个典型的具有网络和用户行为感知的网络视频服务中, 用户获得视频服务, 需要网络提供者所提供的视频内容经过转码、内容管理 (CMS)、内容分发网络 (CDN)、透过不同 ISP 才能到达用户的家里, 这个过程可能经历了多个系统、多个服务商和多个网络. 为了更好的提供视频服务, 服务提供方常常需要分析用户行为、分布以及网络状态和变化趋势, 并通过感知网络状态, 为用户提供合适的服务交付点. 由于网络状态和用户行为随时间变化较大, 这些实时生成的数据类型不同、规模巨大, 而用户对视频服务的请求, 要求系统必须在较短时间内响应并返回合适的服务交付节点. 为此我们采用海云协同方案, 构建了一个对用户行为和网络状态进行采集和实时反馈的网络视频服务测量分析系统, 通过对网络数据的采集和多级别分析, 一方面可为用户选择合适的服务交付点, 另一方面可为服务提供方的服务性能预测、优化和个性化服务提供支撑.

图 4 描述了面向网络视频服务的海云协同分析处理流程, 在该系统中, 首先通过网络爬虫, 将不同视频服务网站的视频服务 URL 进行定期爬取, 通过海服务节点空间中不同的海服务节点访问该内容, 对内容提供方的 CDN 交付点进行分析, 探测网络链路状态, 通过对节点的网络位置进行分析和计算, 可以得到一个视频服务交付点拓扑. 除此之外, 海服务节点以较小的周期采集用户终端的访问记录, 探测与用户端之间的可用带宽、估计用户的网络位置, 并将相关信息进行局部维护. 当用户发起视频服务请求时, 海服务节点根据服务请求端之间的可用带宽、网络连接可靠性、所属 ISP 等信息, 预测可提供服务的交付节点. 同时, 相关信息进行云端汇聚, 对网络内容分布、服务可用性、用户统计行为进行分析和预测, 并反馈给服务提供方, 供其进行内容预部署和服务优化.

网络视频服务分析系统只是海服务和海云协同的一个典型应用. 而在实际的网络中, 许多应用都存在即时响应和全局优化的一个矛盾, 如内容的实时推荐和基于用户历史行为的智能分析的矛盾, 语

图 4 面向网络视频服务的海云协同分析和处理流程

Figure 4 The flowchart of the sea-cloud collaboration for video service in Internet

音图像网络在线识别与高精度模型训练的矛盾, 终端信息采集预处理和后端批量学习的矛盾等, 而海云协同服务可以为这类问题提供一个通用的解决方案. 如何新的架构下对不同问题进行适配、如何针对不同的应用进行系统设计和相关算法的实现, 也是我们未来要研究的问题.

5 总结

本文针对当前网络服务的特征, 以及大量应用对即时响应的需求, 顺应网络大数据处理为主导的服务演变趋势, 采用网络边缘设备为基础资源, 以用户请求为中心, 动态地为其构建一个满足现场服务的弹性自治服务系统 —— 海服务系统. 充分考虑该系统中的网络连接状态、服务请求 QoS 要求、节点计算能力以及网络传输代价等多个因素, 建立了海服务模型, 并研究对应海服务调度方法和系统架构, 是一种新的分布式处理方法, 对互联网即时服务和大数据的高效分级处理, 具有重要的应用价值. 同时, 融合了海服务的局部计算与云计算的全局批量分析, 形成了海云协同服务系统方案. 最后, 以视频服务为例, 构建了基于海云协同的视频服务测量分析系统, 为大数据处理、智能分析和服务以及具有局部、全局融合服务的应用, 提出了新的解决思路.

参考文献

- 1 Jiang M H. Urbanization and information——the developing impportunity of China. JieFang Daily, 2010.10.10 [江绵恒. 城市化与信息化 —— 中国发展的时代机遇. 解放日报, 2010 年 10 月 10 日]
- 2 Buyya R, Yeo C S, Venugopal S, et al. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gener Comput Syst, 2009, 25: 599–616
- 3 Flavio B, Rodolfo M, Jiang Z, et al. Fog computing and its role in the Internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, 2012. 13–15
- 4 Xu Z W. Cloud-sea computing systems: towards thousand-fold improvement in performance per watt for the coming zettabyte era. J Comput Sci Technol, 2014, 29: 177–181
- 5 Hou C D, Li D, Qiu J F, et al. SeaHttp: a resource-oriented protocol to extend REST style for Web of things. J Comput Sci Technol, 2014, 29: 205–215
- 6 Shi H L, Li D, Qiu J F, et al. A task execution framework for cloud-assisted sensor networks. J Comput Sci Technol, 2014, 29: 216–226

- 7 Liu J L, Zhang Y L, Yang L, et al. SAC: exploiting stable set model to enhance cacheFiles. *J Comput Sci Technol*, 2014, 29: 293–302
- 8 Shao B Q, Zhang J W, Zheng C P, et al. A non-forced-write atomic commit protocol for cluster file systems. *J Comput Sci Technol*, 2014, 29: 303–315
- 9 Bao W T, Fu B Z, Chen M Y, et al. A high-performance and cost-efficient interconnection network for high-density servers. *J Comput Sci Technol*, 2014, 29: 281–292
- 10 Chen Y J, Chen T S, Guo Q, et al. An elastic architecture adaptable to millions of application scenarios. In: 9th IFIP International Conference on Network and Parallel Computing Lecture Notes in Computer Science, Gwangju, 2012. 7513: 188–195
- 11 Yin Y X, Chen Y J, Guo Q, et al. Prevention from soft errors via architecture elasticity. *J Comput Sci Technol*, 2014, 29: 247–254
- 12 Huang Z X, Cao F Y, Li J J, et al. Developing sea cloud data system key technologies for large data analysis and mining. *Netw New Media*, 2012, 1: 20–26 [黄哲学, 曹付元, 李俊杰, 等. 面向大数据的海云数据系统关键技术研究. 网络新媒体技术, 2012, 1: 20–26]
- 13 Buyya R, Pathan M, Vakali A. Content Delivery Networks. Berlin: Springer, 2008
- 14 Zhang X Y, Zhang Q, Zhang Z S, et al. A construction of locality-aware overlay network: mOverlay and its performance. *IEEE J Selec Areas Commun*, 2004, 22: 18–28
- 15 Xia Z G, Zhanguo X, Xia S X, et al. A field-based semantic P2P overlay network. In: Proceedings of Wireless Communications, Networking and Mobile Computing, Dalian, 2008. 1–4
- 16 Wang H H, Zhu Y W, Hu Y M. An efficient and secure peer-to-peer overlay network. In: Proceedings of Local Computer Networks, Sydney, 2005. 764–771
- 17 Sun X, Tian Y, Liu Y S, et al. An unstructured P2P network model for efficient resource discovery. In: Proceedings of ICADIWT, Ostrava, 2008. 156–161
- 18 Sun X, Li K, Liu Y S, et al. SLUP: a semantic-based and location-aware unstructured P2P network. In: Proceedings of HPCC, Dalian, 2008. 288–295
- 19 Joung Y J, Lin Z W. On the self-organization of a hybrid peer-to-peer system. *J Netw Comput Appl*, 2010, 33: 183–202
- 20 Chen Z G, Liu J Q, Li D, et al. SOBIE: a novel super-node P2P overlay based on information exchange. *J Comput*, 2009, 4: 853–861
- 21 Koskela T, Harjula E, Kassinen O. Robustness of a P2P community management system based on two-level hierarchical DHT overlays. In: Proceedings of IEEE Symposium on Computers and Communications (ISCC), Kerkyra, 2011. 881–886
- 22 Lima L, Dalai M, Migliorati P, et al. Overlay optimization for peer-to-peer scalable video streaming. In: Proceedings of ICIP, Orlando, 2012. 2245–2248
- 23 Gross C, Stingl D, Richerzhagen B, et al. Geodemlia: a robust peer-to-peer overlay supporting location-based search. In: Proceedings of IEEE P2P, Tarragona, 2012. 25–36
- 24 Baruah S K, Goossens J. Rate-monotonic scheduling on uniform multiprocessors. *IEEE Trans Comput*, 2003, 52: 966–970
- 25 Andersson B, Baruah S, Jonsson J. Static-priority scheduling on multiprocessors. In: Proceedings of RTSS, Washington, 2001. 193–202
- 26 Chetto H, Chetto M. Some results of the earliest deadline scheduling algorithm. *IEEE Trans Softw Eng*, 1989, 15: 1261–1269
- 27 Li L T, Chang H Y, Chao S W. A hybrid task scheduling for multi-core platform, in future generation communication and networking symposia. In: Proceedings of FGCNS, Sanya, 2008. 40–45
- 28 Calandrino J M, Anderson J H, Baumberger D P. A hybrid real-time scheduling approach for large-scale multicore platforms. In: Proceedings of ECRTS, Pisa, 2007. 247–258
- 29 Anderson J H, Calandrino J M. Parallel real-time task scheduling on multicore platforms. In: Proceedings of RTSS, Rio de Janeiro, 2006. 89–100
- 30 Baruah S, Funk S, Goossens J. Robustness results concerning EDF scheduling upon uniform multiprocessors. *IEEE Trans Comput*, 2003, 52: 1185–1195
- 31 Baruah S K, Cohen N K, Plaxton C G, et al. Proportionate progress: a notion of fairness in resource allocation.

Algorithmica, 1996, 15: 600–625

- 32 Anderson J H, Srinivasan A. Mixed Pfair/ERfair scheduling of asynchronous periodic tasks. In: Proceedings of 13th Euromicro Conference on Real-Time Systems, Delft, 2001. 76–85

Research and design of an on-site, elastic, autonomous network— —SEA Service System

WANG JinLin^{1*}, TIAN Jing², YOU JiaLi¹, LIU Xue¹ & DENG HaoJiang¹

1 *National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China;*

2 *Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

*E-mail: wangjl@dsp.ac.cn

Abstract According to the developing Internet trend of focusing mainly on the user, we propose an on-site, elastic, and autonomous network, the SEA Service Network, which can provide real-time services and guarantee the QoS of applications. In this network, the request is the center of the service, and the system provides an appropriate service node for different requests based on several factors such as network characteristics, QoS constraints, and service type. In addition, considering the features of SEA and Cloud services, we combine these two service modes in a SEA-Cloud collaboration system to provide a new architecture for big data processing. In this architecture, local information can be combined easily and processed real-time by the SEA service part while global information is collected and analyzed periodically by the Cloud service. This architecture can meet different QoS or model training requirements and provide a more efficient processing mechanism for network service.

Keywords SEA service, real-time service, service scheduling, SEA-cloud collaboration, data processing

WANG JinLin is Professor at the Institute of Acoustics at the Chinese Academy of Sciences. Currently, he is Director of the Network and New Media Technology Research Center. His current research interests include structures and new service with broadband networks, digital media services, and network architectures.