

基于服务等级的网格服务质量描述语言^{*})

唐亚哲 曹宁 沈琳 张凯 李亚

(西安交通大学计算机科学系 西安 710049)

摘要 针对网格服务质量管理中用服务水平协议(SLA)带来较大协商开销的问题,提出基于服务等级的网格服务质量描述语言即网格服务等级语言(CGSL)及建立在CGSL之上的网格服务质量管理框架。服务等级是轻量级SLA,提供一种区分不同服务质量的机制。CGSL基于XML技术和WSRF规范,与Web服务描述语言(WSDL)兼容,用于描述网格服务提供方所能提供的不同服务等级,为监控服务质量提供具体信息。网格服务质量管理框架建立在CGSL之上,并通过Mobile Agent监控服务质量是否得到满足。一个具体的文件存储网格服务实例用来展示CGSL在网格服务质量管理中的应用。该实例表明CGSL以少量的服务等级代替复杂的SLA,能有效降低运行时消耗,适合异构多变的网格环境。

关键词 网格服务,服务质量,服务等级,网格服务等级语言

CGSL-class of Grid Services Language

TANG Ya-zhe CAO Ning SHEN Lin ZHANG Kai LI Ya

(Department of Computer Science, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract To reduce the negotiating cost brought by the Service Level Agreement(SLA) in the management of qualities of grid services, this paper proposes the Class of Grid Service Language (CGSL) and its corresponding QoS Management Framework for Grid Services on the basis of Class of Service (CoS). As a lightweight SLA, CoS can distinguish different qualities of services (QoS). Based on Extensible Markup Language (XML) and Web Services Resource Framework (WSRF), CGSL is compatible with Web Service Definition Language (WSDL). It is used to offer detailed information about the qualities of grid services as well as to describe different CoSs supported by service providers. The corresponding QoS Management Framework is founded on both the CGSL and the Mobile Agent which monitors the qualities of grid services. A storage grid service is presented to show how to manage the qualities of grid services with the help of CGSL. The result shows that CGSL effectively reduces the runtime overhead by using a few CoSs instead of complex SLAs, therefore is suitable for the heterogeneous and inconstant grid environment.

Keywords Grid service, QoS, Class of service, CGSL

1 前言

网格服务最重要的目标就是整合分布式异构资源,向用户提供无缝的高质量服务。QoS(Quality of Service, 服务质量)管理涉及两方面的信息:一是约束信息,描述服务质量应该达到的精度,如响应时间不得超过10ms、下载速度不得低于200kB/s等;二是管理信息,描述对Web服务进行管理所需的信息,如某种服务的使用费等。

在针对Web服务的WSLA^[1]中,每个用户都可以和服务提供方签订SLA(Service Level Agreement, 服务水平协议)文档,从而在一个最优的服务水平消费服务。但是在网格环境中存在大量临时性的服务提供方和消费方,若采取逐个签订SLA的做法,不但会增加运行时消耗,也会降低暂时性服务的灵活性。基于此种考虑,我们提出了基于Mobile Agent的网格监控系统MAGMS^[2]和基于服务等级的网格服务质量管理框架CQEF^[3,4],MAGMS将SLA简化为服务等级(Class of Service)并将其引入到网格服务中,提出网格服务等

级(Class of Grid Services)的概念。服务等级作为一种轻量级的SLA存在于服务双方之间,服务端提供若干个固定的服务等级,客户端根据消费能力和服务质量从中选择一个最合适的服务等级,服务端按照客户端所选择的服务等级提供服务。这种协商过程以少量的服务等级代替了复杂多样的SLA,有效降低运行时消耗,在复杂多变的网格环境下效果尤其明显。

本文提出的网格服务等级语言CGSL(Class of Grid Services Language),用于描述网格服务的不同服务等级(CoS)。每个CoS由不同的约束(Constraint)构成,约束是对具体资源的属性要求或对网格服务的要求,例如对网络带宽的要求、对网格服务响应时间的要求、对内存空间的要求等。

在基于CGSL的网格服务质量监控框架中,网格服务提供方注册服务时不但要提供一个WSDL文档来描述此服务提供的功能及访问方式,还需提供一个CGSL文档来描述此服务所能提供的不同服务等级。此外,我们在此监控框架中设计了监控模式库^[2],它为系统提供QoS参数的数据采集方

^{*}基金项目:国家自然科学基金(No. 60403029)。唐亚哲 副教授,研究方向为网络管理与分布式系统;曹宁 硕士研究生,研究方向为Mobile Agent与网格服务质量管理;沈琳 硕士研究生,研究方向为网络资源管理;张凯 硕士研究生,研究方向为网格服务评估;李亚 硕士,研究方向为网格服务监控。

法,将每种 QoS 参数的数据采集方法组装成一个监控模块。当用户请求某个具体的网格服务时,系统解析 CGSL 文档并显示不同的服务等级,用户将根据自己的需求选择某个服务等级。系统根据该服务等级中约束所涉及的 QoS 参数,在监控模式库中查找与该参数相关的监控模块。所有涉及到的监控模块组装成具有监控逻辑的监控实体即 Mobile Agent,以此来管理网格服务的质量。

本文组织结构如下:第 2 节介绍相关工作,第 3 节阐述 CGSL 的组织结构,第 4 节描述基于 CGSL 的网格服务质量管理框架,第 5 节通过网格服务实例展示 CGSL 的应用,最后总结 CGSL 的优缺点以及后续工作。

2 相关工作

WSDL(Web Service Description Language, Web 服务描述语言)^[5]是用 XML 文档来描述 Web 服务的标准,是 Web 服务的接口定义语言。WSDL 可描述 Web 服务的三个基本属性:服务所提供的操作(方法);和服务交互的数据格式以及必要协议;协议相关的地址,如 URL。WSDL 文档以端口集合(PortTypes)的形式来描述 Web 服务,WSDL 服务描述包含对一组操作(Operation)和消息(Message)的抽象定义、绑定到这些操作和消息的一个具体协议和这个绑定的一个网络端点规范。但是 WSDL 只对服务的功能做出描述,并未涉及服务质量的管理。

IBM 提出的 WSLA^[6]和惠普开发的 WSML^[7]都是基于服务双方协商的 SLA 语言,并提供相应的 Web 服务管理基础设施 WSLA Compliance Monitor^[1]。SLA 包含 QoS 保证和管理信息(比如管理职责、价格),其中 QoS 保证是以服务水平对象(SLO)形式提供。WSLA 和 WSML 不支持服务等级,其服务签订过程是基于服务消费方与提供方进行协商的主动方式,而 CGSL 采用基于消费方选择服务等级的被动方式,两种方式引起的时间开销有很大差别。此外,无论是 WSLA 还是 WSML 都不支持功能约束。

WSML 在外部库定义度量,需第三方语言(比如 MathML)的支持,由此会引起兼容性问题。而 CGSL 直接使用内部元素(CGSL 参数)来定义 QoS 度量,不仅避免了兼容性问题,而且便于监控系统(如 Mobile Agent)的快速生成。WSLA 和 WSML 都面向企业的管理应用,都有强大复杂的管理基础设施,但这些管理基础设施并不适应异构多变的网格环境。从整体性能来看,WSLA 和 WSML 的运行开销比简化的 CGSL 要高,轻量级的 CGSL 及其网格服务质量管理框架更适合对运行开销特别敏感的网格环境。

Vladimir Tomic 等人提出的 WSOL(Web Service Offering Language)^[8]是基于 XML 的服务等级语言,它指定一个 Web 服务具有一个或多个服务等级。在 WSOL 里,Service Offering 表示为 Web 服务的一个服务等级。WSOL 描述和规范 Web 服务的等级,还支持 Web 服务的组合管理,WSOL 的基础设施 WSOI 被用来管理和监控服务等级。此外,WSOL 并没有提供与其相关联的 WSDL 文档的位置信息,也没有定义 QoS 的度量。而 CGSL 通过定义 QoS 度量以及 QoS 参数集,大幅提高约束的多样性,使得用户可以从多角度多层次来评价服务质量。

由于网格服务和 Web 服务的差异,为 Web 服务设计的 WSLA,WSML 和 WSOL 等服务管理语言都不能直接支持网格服务应用。在文献^[9]中给出了移植 WSLA 到网格环境下

的方法,但 SLA 不能支持 CoS。此前尚未有网格环境的服务等级描述语言,而 CGSL 则专门为网格服务设计,支持运行在有状态资源之上的网格服务。CGSL 作为轻量级的网格服务等级语言,继承 WSOL 提出的基础概念(如服务等级),并针对网格环境的自身特点添加了相关概念(比如 QoS 参数集、CGSL 参数等)。

3 CGSL

CGSL 作为对 WSDL 的扩展,不仅要与 WSDL 兼容,还要简洁、清晰并规范地描述网格服务等级。CGSL 的主要概念包括服务等级(CoS)、约束(Constraint)、约束组(Constraint Group,简称 CG)、CGSL 参数(CGSL Parameter)和管理信息(Management)等。同一个服务可以提供多个等级的服务质量,因此与服务所关联的 CGSL 文档可包含若干个服务等级。一个服务等级包括若干个约束和约束组,约束是一组对 CGSL 参数进行评价的表达式。Description 元素用于描述参数的基本信息,Metric 元素定义该参数的测量和计算方法。约束组由若干个约束及约束组构成,不同的约束组可以共享各自定义的约束和约束组。管理信息主要包括服务约束管理信息和服务等级管理信息。CGSL 中主要概念及其相互关系如图 1 所示。

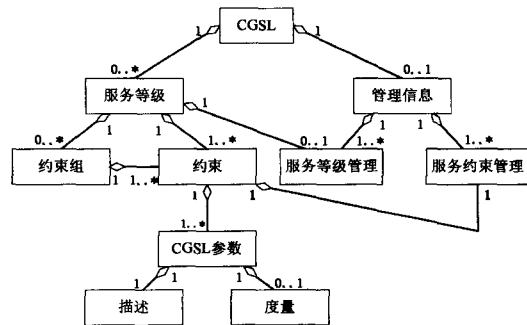


图 1 CGSL 主要概念

服务等级是 CGSL 里最重要的概念,包括约束信息和管理信息。它由若干个约束和约束组构成,用于描述不同等级的服务所提供的具体服务质量。此外,服务等级也可包含服务等级管理信息,主要指明用户使用该服务所支付的报酬,以及服务提供方无法满足该等级的服务质量时所需缴纳的罚金或其他惩罚措施。

约束是对 CGSL 参数的评价,比如“响应时间小于 50ms”。CGSL 用逻辑表达式来描述约束,对约束做评价可以在调用操作之前、调用操作之后或在特定时间间隔后。根据不同的约束对象,我们把约束分为功能约束和 QoS 约束。功能约束主要用于规定用户使用网格服务所提供的操作时双方应满足的条件,QoS 约束则规定了与服务质量直接相关的 QoS 参数所应达到的水平。我们依据不同的取值方式将 QoS 参数分为原子参数和非原子参数两种。原子参数是指可由监控模式库中相应的监控模块直接获得其值的 QoS 参数,例如在监控模式库中有与服务执行时间参数直接对应的 Proceed-Time 监控模块,此模块封装了服务执行时间参数的取值算法,即在服务提供方用发送服务响应的时刻减去收到服务请求的时刻;非原子参数则无法由某一监控模块直接计算其值,例如监控模式库中没有与平均执行时间参数直接对应的监控模块,如果某一服务等级包含对此参数的约束,那么相应的

CGSL 文档中必须明确表述此参数的计算方法。QoS 约束根据 QoS 参数的类型相应分为两种,针对原子参数的约束,称为原子 QoS 约束;针对非原子参数的约束,则称为非原子 QoS 约束。

功能约束规定了调用某种操作须满足的条件。比如调用数学服务的除法操作时,需检查除数是否非零。CGSL 支持前置条件、后置条件、预置条件和恒量条件等四种检查方式。前置条件要求检查用户调用服务的参数或(和)服务提供方的最初状态是否有效;后置条件要求检查服务提供方提供的服务结果是否正确以及服务提供方的最终状态是否有效;预置条件要求在服务提供方调用一个操作后,监控方经过指定时间间隔,检查服务提供方发送给用户的结果是否正确;恒量条件要求检查服务的操作在调用前和调用后都被满足,可以由前置条件和后置条件复合而成。功能约束的检查位置一般在服务提供方或用户方。

原子 QoS 约束指定了原子 QoS 参数应满足的阈值,一个特定的操作调用可能需要评价许多原子 QoS 约束。原子 QoS 参数描述了服务的执行性能、可访问性或可靠性等属性。原子 QoS 约束说明需要监控的 QoS 内容,并指明监控模式库中与之对应的监控实体模块和监控时间。比如,要取得某一操作的执行时间,必须指出监控模式库中相应的执行时间监控模块。其中,周期性原子 QoS 约束是一类特殊的原子 QoS 约束,它需要按指定时间间隔做周期性评价,可通过设置原子 QoS 约束的 Method 属性来指定时间间隔。比如,设置 Method 的 PeriodCondition 元素为 4,则每 4 次操作调用就进行 1 次约束检查。此外,若对每次调用都做约束评价,则产生的开销较大。为减小此类开销,CGSL 支持随机地对每种操作的部分调用做 QoS 约束评价,即抽样约束。CGSL 的抽样约束以牺牲管理信息的质量和精确度为代价,不过这种代价在特定环境下是值得的。为最大限度地提高抽样约束评价的有用性,服务提供方并不知道哪次调用要做抽样约束评价,这就避免了服务提供方有针对性地提高某次调用的服务质量。原子 QoS 约束的抽样评价一般由第三方完成。

对于用户来说,有的网络服务应该首先保证其稳定性,有的网络服务应该尽量减少其响应时间。用户使用不同类型的网络服务所关注的服务质量不尽相同,服务提供方也应针对网络服务的类型从不同的角度对服务质量给以表述。然而原子 QoS 约束所提供的服务质量表述方式较为单一,它只能从原子 QoS 参数的角度对服务质量进行表述,远远不能满足上述情况下对于表述方式多样化的需求。例如,网络服务提供方为突出其网络带宽的优势而打算对“网络传输时延”参数作出约束,但是监控模式库中并没有对应此参数的监控模块,即此参数属于非原子 QoS 参数,不能通过原子 QoS 约束来描述此方面的服务质量。然而监控模式库含有“服务执行时间”和“服务响应时间”两个相关的原子 QoS 参数的监控模块,利用这两个参数值便可以计算非原子 QoS 参数“网络传输时延”的值。

基于上述非原子 QoS 参数的评价考虑,CGSL 设计了非原子 QoS 约束。与原子 QoS 约束不同的是,非原子 QoS 约束不但要描述非原子 QoS 参数所满足的阈值条件,还要表述此参数的测量和计算方法。由于非原子 QoS 参数不能由监控模式库中的监控模块直接计算,一般通过对原子 QoS 参数的组合运算来计算其值,CGSL 设计了专门的度量(Metric)元素来描述求值算法等信息。比如,针对非原子 QoS 约束“网

络传输时延不得大于 500ms”,服务提供方不但要描述非原子 QoS 参数“网络传输时延”的上限是 500ms,还需通过元素 Metric 描述此参数的计算方法即“响应时间减去服务执行时间”。非原子 QoS 约束的评价可以在任何一方(提供方、消费方、第三方)进行,它的引入丰富了服务质量的描述方式,使得服务提供方可以从多个角度灵活描述自己的服务质量。

CGSL 基于最新的 Web 服务资源框架 WSRF^[10],网络服务不同于一般的无状态 Web 服务,它是运行在有状态资源之上的服务。因此,在评价其服务质量时不仅要考虑短时间内单次服务的质量,还要综合考虑长时间内多次服务的质量。CGSL 通过基于 QoS 参数集的非原子 QoS 约束来提供这种一段时间内的整体性服务质量评价。QoS 参数集指一段时间内满足一定条件的某一种参数值的集合,CGSL 允许服务方在此集合上定义非原子 QoS 约束。比如非原子 QoS 约束“30s 内服务响应时间超过 200ms 的次数不超过 10 次”,它将最近 30s 内超过 200ms 的所有响应时间参数值组成 QoS 参数集,并且对此集合做计数(Count)操作,使用户得以从最近一段时间内的响应时间角度来评价此网络服务。结合 QoS 参数集的约束能使服务提供方突出自己在一段时间内整体服务质量的优点,也使用户从更高的层次来获得服务质量的保障。

一般来说,同一个服务的不同等级所拥有的约束不完全一样,但也能拥有一些相同的约束。例如,服务等级 COS1 和服务等级 COS2 对于响应时间和故障次数的约束不同,而对于带宽参数的约束相同。如果 COS1 和 COS2 都各自定义对带宽的约束,则完全相同的带宽约束将在 CGSL 文档中被定义两次。为避免重复定义相同的约束,我们在 CGSL 中引入了约束组。约束组是个嵌套概念,它由若干个约束和(或)约束组构成;它既可以在内部定义新的约束或约束组,也可以引用本 CGSL 文档中在其他位置定义的约束或约束组。此外,约束组还可继承已存在的约束或约束组,并可以新增约束或约束组的定义。引入约束组的最大目的就是提高约束的可重用性,不同的服务等级可以共用相同的约束组,而且不同的约束组也可以共用相同的约束和管理信息。若一个网络服务提供较多的服务等级,不同服务等级的差异较少,那么拥有相同约束的可能性很大,此时引入约束组将会大大降低为网络服务开发 CGSL 文档的工作量。

WSDL 语言不支持服务管理,因此 CGSL 针对网络服务的管理必不可少。CGSL 中针对网络服务的管理包括服务等级的管理和约束的管理。服务等级的管理主要涉及服务提供方同用户之间的业务信息,如服务价格和违约金。服务等级不同,相应的价格和违约金也不同。基本原则是服务质量越好,价格越高,违约金也越高。此外,服务等级的有效期也属于管理信息。服务约束的管理主要涉及约束的类型信息及检查方法。约束分为功能约束、原子 QoS 约束和非原子 QoS 约束等三种类型;相应的检查方法也不同;功能约束支持前置条件等四种检查方式,原子 QoS 约束和非原子 QoS 约束则支持周期检查和抽样检查两种检查方式。针对服务等级的管理信息为服务等级管理对象(ServiceCoSObject),而针对约束的管理信息为服务约束管理对象(ServiceConstraintObject)。

Web 服务技术是网络服务的关键支撑技术,而 Web 服务技术的基础是 XML。因此,CGSL 同样基于 XML 技术,采用 XML Schema 定义自己的文法。CGSL 文法结构如图 2 所

示。

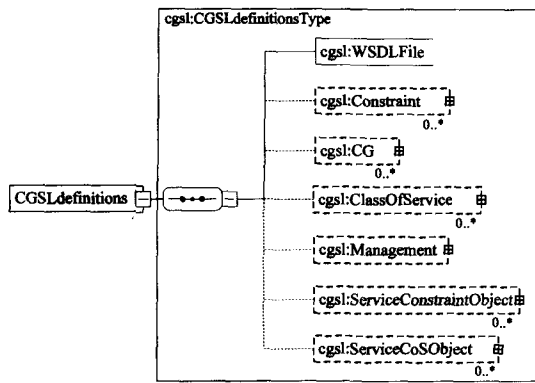


图2 CGSL 基本文法结构图

CGSL 采用单根结构,根元素 CGSLdefinitions 定义 CGSL 的基本结构,主要分为三个部分:第一部分是同 CGSL 文档相关联的 WSDL 文档位置;第二个部分是关于服务等级的定义信息,包括约束 Constraint、约束组 CG 和服务等级 ClassOfService;第三个部分是服务的管理信息 Management、服务约束管理对象 ServiceConstraintObject、服务等级管理对象 ServiceCoSObject。

CGSL 的文法定义参考了 WSOL 的文法设计,但两者有较大区别。首先,CGSL 定义了 WSDL 文档关联位置的元素,而 WSOL 则没有相关信息。在 CGSL 里,约束的属性信息由 CGSLParameter 定义,CGSLParameter 用 Description 描述 CGSL 参数,用 Metric 定义如何测量 CGSL 参数;WSOL 的约束属性由它专门设计的 Expression Schema 来定义,Expression Schema 负责描述 WSOL 参数,有专门的本体文档负责其语义的理解,WSOL 没有定义度量。在 CGSL 中用 Management 元素定义管理信息,而 WSOL 选择 statement 元素;CGSL 有且只有一个 Management 元素,它包含若干个 ServiceConstraintObject 和 ServiceCoSObject,WSOL 的 statement 可以跟 Constraint、CG 组合出现若干次;CGSL 用 ServiceCoSObject 元素定义服务等级的有效期限,WSOL 没有这方面的内容。

```
<cgsl:Constraint name="C3" service="Storage:StorageService"
portOrPortType="Storage:StoragePortType" operation="CGSL-ALL">
  <Expression>
    <cgsl:CGSLParameter name="AveResponseTime" type="float" unit="ms">
      <cgsl:Description>
        <cgsl:ParameterName>AveResponseTime</cgsl:ParameterName>
        <cgsl:Op>Less</cgsl:Op>
        <cgsl:Value>100</cgsl:Value>
      </cgsl:Description>
      <Metric>
        <Operand>
          <cgsl:Function name="Ave">
            <cgsl:Select unit="ms" type="float">ResponseTime</cgsl:Select>
          </cgsl:Function>
        </Operand>
      </Metric>
    </cgsl:CGSLParameter>
  </Expression>
</cgsl:Constraint>
```

图3 非原子 QoS 约束的 CGSL 描述示例

图3 给出非原子 QoS 约束“平均响应时间应该小于 100ms”的 CGSL 定义。首先用 Description 元素定义了 CGSL 参数 AveResponseTime 应满足的阈值条件,即平均响应时间应该小于 100ms;然后用 Metric 元素描述了此参数的计算方法,即将所有的 ResponseTime 值组成参数集,然后对此集合求平均值。此处并没有进一步表明 ResponseTime 的计算方

法,因为 ResponseTime 是原子 QoS 参数,在监控模式库中有专门与之对应的监控模块提供它的计算方法。监控模式库如同编程语言中的标准函数库,监控模块则像标准函数一样只提供基本 QoS 参数(即原子 QoS 参数)计算方法,CGSL 使得我们可以通过调用这些监控模块来计算扩展 QoS 参数(即非原子 QoS 参数),利用这种机制可以非常灵活地表述网络服务质量,并且也不会增加监控模式库的复杂度,还可避免整个监控框架的过于庞大。

CGSL 是用 XML Schema 描述的 XML 语言,为其提供支持和管理的语言工具必不可少,因此我们开发了 CGSL 解析器的原型。CGSL 解析器用来解析 CGSL 文档的应用程序,它检查和报告 CGSL 文档的语法和语义错误,帮助开发者撰写符合 CGSL 语法规则的网络服务等级描述文档。目前 XML 解析器有很多,比如 XML Parser for Java, Microsoft XML Parser (MSXML), Xerces Java Parser, Xerces2 Java Parser, Xerces C++ Parser, Oracle's XML Parser for Java 等。其中,Xerces2-java^[11]是 Apache 用 java 开发的开源 XML 解析器,具有良好的跨平台能力,且方便进行扩展。此外,Xerces2-java 不但支持标准的 DTD 及 XML Schema,而且它的 XNI 框架提供简易的添加组件以及扩展组件的途径。其它解析器都不同时具备上述的特点(高性能、开源、跨平台、易于扩展,支持标准的 DTD 及 XML Schema),所以我们选择 Apache 的 Xerces2-java 作为基础,并对其扩展来实现 CGSL 解析器。

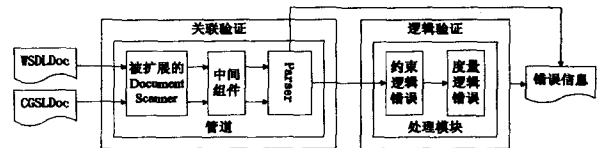


图4 CGSL 解析流程图

整个解析验证流程如图4所示,主要分为关联验证和逻辑验证两个部分。我们在基于 Xerces2-java 的多 XML 文档关联验证^[12]中详细描述了关联验证的内容,它主要检验 CGSL 文档中多个元素之间相互引用的正确性,可以检查两类错误:一是外关联错误,CGSL 文档引用 WSDL 文档所声明的元素时有可能出现此类错误;二是内关联错误,比如在 CGSL 文档中元素 ServiceCoSObject(即服务等级管理对象)对元素 CoS(即服务等级)的引用。逻辑验证主要检验 CGSL 文档的内部逻辑语义错误,比如约束的上限小于下限等此类错误。关联验证和逻辑验证产生的错误信息被送到错误报告文件后,CGSL 文档的整个解析验证流程完成。

4 基于 CGSL 的网络服务质量管理框架

基于 CGSL 的网络服务质量管理框架主要采用了 CGSL 和 Mobile Agent^[13]两大关键技术。其中 CGSL 用于描述网络服务的等级,可以大幅度减少服务双方的协商开销;Mobile Agent 负责监控网络服务的等级是否符合相应 CGSL 中描述的服务等级,能够降低由服务质量管理本身带来的网络负载。此管理框架将网络中所有节点分为四类:注册节点、监控节点、用户节点和服务节点。基于 CGSL 的网络服务质量管理框架结构如图5所示。

我们通过扩展现有的 UDDI^[14]构建网络服务的注册节点即 UDDIx^[15],使之成为支持服务等级的网络服务注册和查询

中心。UDDI(通用发现与发布方法集)规范定义了一种发布和发现 Web 服务信息的方式,对其实现细节没有进行严格的规定,也没有提出参考的实现规范。目前版本的 UDDI 规范更多地关注于基于功能约束的服务发现问题,并没有考虑服务质量的内容。为克服这些缺点,需要对现有的 UDDI 模型进行必要的扩展从而实现服务质量感知的注册及发现过程。我们在服务注册及发现过程中采用服务等级 CoS 来体现服务质量 QoS,首先服务提供方在注册服务时需提供该服务相对应的 CGSL 文件,该 CGSL 文件描述了该服务所能提供的若干服务等级,每个服务等级可能在各自的服务质量或者代价上有所区别。当用户从若干个具有相同功能的服务中选取一个的时候,他们可参考这些服务的 QoS 描述文件(即 CGSL 文档)。

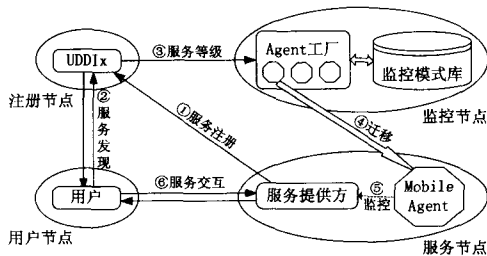


图5 基于CGSL的网络服务质量管理框架

监控节点是整个服务质量管理框架的核心节点,由 agent 工厂^[2]和监控模式库组成。Agent 工厂负责产生用于监控网络服务的 Mobile Agent,监控模式库由提供各种 QoS 参数采集方法的监控模块组成。所有在 CGSL 中出现的 QoS 参数都必须在监控模式库里面定义,由监控模式库对应的监控模块提供该参数的数据采集方法等信息。监控模式库可以监控的参数包括针对资源属性的参数(如带宽的参数),以及针对网络服务的参数(如响应时间的参数)。WSRF 规定可以通过标准的 Web 服务来读取、修改及查询有状态资源的属性,因此监控模式库可利用此方法来进行 QoS 参数的数据采集,此外还可利用文件读取(如通过 UNIX 系统的 /proc 虚拟文件系统获取服务提供方的系统运行时信息)、SNMP 协议等一切可以采集数据的方法。Agent 工厂根据服务等级信息把相关的监控模块组合在一起,生成一个具有移动性和自主性的 Mobile Agent,并将其迁移至网络服务节点对网络服务进行服务质量监控。

用户节点是为用户提供使用网络服务的平台,用户通过此节点选择并使用网络服务。服务节点不但为网络服务提供运行平台,还为 Mobile Agent 提供一个运行环境,包括 Mobile Agent 的启动、挂起、唤醒以及与外界交互等。

在网络服务质量管理框架中,各个组件之间的交互流程如下。

(1)服务注册。服务提供方在 UDDIx 中注册服务,需要提供描述其服务功能的 WSDL 文件和描述此服务上所有服务等级的 CGSL 文件。

(2)服务发现。用户查询 UDDIx 的服务分类目录信息,并根据自己的需求选定某一类;然后 UDDIx 将所有此类服务的 WSDL 文件及其 CGSL 文件返回给用户;用户节点通过解析 CGSL 文件获知各服务所能提供的服务等级。这样用户在选择服务的时候不仅可以考虑服务的功能,还能考虑以服务等级形式体现的服务质量。

(3)监控准备。用户选定网络服务及服务等级后,UDDIx 将相应的服务等级信息(CGSL 文件)传至监控节点。Agent 工厂解析此 CGSL 文件,找到相关约束条件并提取需要监控的参数和逻辑;从监控模式库中找到对应参数的监控模块,根据监控逻辑将监控模块进行封装,最终生成 Mobile Agent。

(4)Mobile Agent 迁移。网格中的信息交互都采用 SOAP 消息的形式,所以 Mobile Agent 通过 SOAP 通道迁移至待监控的服务节点,随后在服务节点进行注册以获取运行环境。

(5)服务交互。Mobile Agent 在服务节点做好监控准备后,给用户发送准备完毕的通知,随后用户绑定并调用该服务。

(6)服务监控。一旦用户开始调用网络服务,Mobile Agent 将持续对此服务的交互过程进行服务质量监控;若服务质量不能满足相应的服务等级,Mobile Agent 通知服务提供方以便使其及时作出调整,还将通知用户使其可以索取罚金或更改服务提供方。

5 CGSL 应用实例——文件存储网格服务

文件存储网格服务提供若干个服务等级供用户选择使用,本节以此实例说明如何用 CGSL 描述网络服务的服务质量需求。

文件存储网格服务基于 Globus Toolkit4^[16]开发,向用户提供文件存储服务。文件存储网格服务是文件存储在网格环境的应用,是在网格环境下模拟网络文件存储的过程。此服务包括用户登录、上传文档、下载文档、查看文档大小、查看文档列表、获取总存储空间大小、获取空闲空间大小以及获取使用空间大小等功能。根据文件存储服务的功能需求,服务提供方用 WSDL 文档描述文件存储服务的功能接口,篇幅有限,在此不予描述。文件存储网格服务提供方根据自己的 QoS 担保能力和用户关心的服务质量,提供三个服务等级供用户选择。

等级 A:故障次数低于 10 次;上传速度和下载速度不得低于 10k/s;请求响应的平均时间小于 500ms;该等级的有效期 1 年;价格是每月付 2 美金,罚金是服务提供方违约一次需支付 1 美金。

等级 B:故障次数低于 10 次;上传速度和下载速度不得低于 40k/s;请求响应的平均时间小于 500ms;该等级的有效期 1 年;价格是每月付 6 美金,罚金是服务提供方违约一次需支付 2 美金。

等级 C:故障次数低于 5 次;上传速度和下载速度不得低于 40k/s;请求响应的平均时间小于 200ms;该等级的有效期 1 年;价格是每月付 10 美金,罚金是服务提供方违约一次需支付 5 美金。

为使用 CGSL 语言描述文件存储服务的等级信息,先从服务等级描述信息里抽象出三个 CGSL 参数:故障次数 Downtime、平均响应时间 AveResponseTime 和网络带宽 NetSpeed。

服务等级 A 的定义如图 6 所示。服务等级 B 和 C 的定义与其类似,故不在此赘述。

在故障次数 Downtime 的约束 C1 定义中,属性 service 指明它是针对 StorageService 服务的约束,元素 Description 指明此约束对 QoS 参数 Downtime 的阈值要求,即小于 10 次。由于此约束是非原子 QoS 约束,不能直接根据监控模式库得

到 Downtime 的测量方法,因此需要通过元素 Metric 描述 Downtime 的计算方式,即统计响应时间大于 5s 的次数。平均响应时间 AveResponseTime 的约束 C3 也是非原子 QoS 约束,定义方法类似 C1,在此不予赘述。网络带宽 NetSpeed 的约束 C5 是非原子 QoS 约束,CGSL 参数 NetSpeed 可以通过监控模式库获得测量方法,在此只需说明需满足的阈值要求,即大于等于 10k/s。

```
<ClassOfService name="CoS-A" service="Storage:StorageService">
  <cgsl:Constraint name="C1" service="Storage:StorageService" portOrPortType="Storage:
    StoragePortType" operation="CGSL-ALL">
    <Expression>
      <cgsl:CGSLParameter name="Downtime" type="integer" unit="nums">
        <cgsl:Description>
          <cgsl:ParameterName>Downtime</cgsl:ParameterName>
          <cgsl:Op>Less</cgsl:Op>
          <cgsl:Value>10</cgsl:Value>
        </cgsl:Description>
        <Metric>
          <Operand>
            <cgsl:Function name="Count">
              <cgsl:Select type="float" unit="seconds">ResponseTime</cgsl:Select>
              <Where>
                <cgsl:ParameterName>ResponseTime</cgsl:ParameterName>
                <cgsl:Op>Greater</cgsl:Op>
                <cgsl:Value>5</cgsl:Value>
              </Where>
            </cgsl:Function>
          </Operand>
        </Metric>
      </cgsl:CGSLParameter>
    </Expression>
  </cgsl:Constraint>
  <cgsl:Constraint name="C3" service="Storage:StorageService" portOrPortType="Storage:
    StoragePortType" operation="CGSL-ALL">
    ...
  </cgsl:Constraint>
  <cgsl:Constraint name="C5" service="Storage:StorageService" portOrPortType="Storage:
    StoragePortType" operation="uploadDataFile">
    <Expression>
      <cgsl:CGSLParameter name="NetSpeed" type="float" unit="KPerSecond">
        <cgsl:Description>
          <cgsl:ParameterName>NetSpeed</cgsl:ParameterName>
          <cgsl:Op>GreaterEqual</cgsl:Op>
          <cgsl:Value>10</cgsl:Value>
        </cgsl:Description>
      </cgsl:CGSLParameter>
    </Expression>
  </Constraint>
</ClassOfService>
```

图 6 网格服务等级的 CGSL 定义示例

对于用户来说,在查询此文件存储网格服务时,由于服务提供方给定的三个服务等级已经规定了具体的服务质量,因此不需要再和服务提供方就服务质量的详细内容(如故障次数、响应时间等参数)进行协商,只需在三个服务等级中加以选择即可;对于服务提供方来说,当有大量用户访问此文件存储网格服务时,服务提供方无需逐一和单个用户就服务质量的每个参数进行协商,而只需将三种服务等级提供给用户即可。这种机制以服务等级的简单选择代替服务质量的复杂协商,大大减少用户和服务提供方在服务发现阶段的握手协商时间,从而降低整个服务质量管理给网格服务带来的额外开销。

结束语 本文将服务等级引入网格服务环境,并提出网格服务等级语言 CGSL。该语言用来描述网格服务的质量信息和管理信息,主要包括约束、约束组、服务等级及管理信息等概念。它借鉴 WSLA 和 WSOL 的优点,能够准确简洁地描述网格服务的等级,以少量的服务等级代替复杂多样的 SLA,有效降低服务双方的协商时间,利于服务质量管理的实施。

CGSL 参数体系基于监控模式库建立,任何出现在 CGSL 中的原子 QoS 参数都已在监控模式库里面定义,由监控模式库负责提供该参数的相关信息,CGSL 参数与监控模式库的结合为获取 QoS 参数值提供了一种简单快捷的方式。此外,CGSL 基于 WSRF 框架,因此支持更丰富的 QoS 需求描述,比如基于 QoS 参数集的非原子 QoS 约束,大幅提高约束的多样性,使得用户可以从多角度多层次来评价服务质量。

网格服务等级语言与 WSDL 兼容,网格服务提供方可自

由选择是否提供 CGSL 文件。网格服务等级语言为网格环境设计,为解决网格环境下基于服务等级的服务质量管理奠定基础。CGSL 目前还需要进一步完善,如 CGSL 解析器的完善、对网格服务组合管理的支持以及相关的开发工具等,这也是我们下一步的工作重点。

参考文献

- [1] Keller A, Ludwig H. The WSLA Framework; Specifying and Monitoring Service Level Agreements for Web Services [R]. IBM Research Report, International Business Machines Corporation (IBM), May 2002
- [2] Chen A, Tang Y, Liu Y, et al. MAGMS; Mobile Agent-based Grid Monitoring System [C] // Proc. of APWeb2006 (Harbin, China, January 2006). LNCS3841, 2006; 739-734
- [3] Tang Yazhe, Yang Yubin, Zhao Ming, et al. CoS-based QoS management framework for grid services // The 6th International Conference on Grid and Cooperative Computing. Urumchi, Xinjiang, China, 2007; 16-18
- [4] 杨宇彬,唐亚哲,姚雷,等.基于 Wrapper 和 WSRF 的服务等级监控机制 [J]. 计算机工程, 2007, 33(7): 109-111
- [5] Web Services Description Language (WSDL) 1.1 [EB/OL]. http://www.w3.org/TR/wsdl (2006-3-8)
- [6] Ludwig H, Keller A, Dan A, et al. Web Service Level Agreement (WSLA) Language Specification [EB/OL]. Ver. 1.0, Revision wsla-2003/01/28. Int. Business Machines Corporation. http://www.research.ibm.com/wsla/WSLA_SpecV1-20030128.pdf (2006-4-15)
- [7] Sahai A, Durante A, Machiraju V. Towards Automated SLA Management for Web Services. HP Laboratories, Palo Alto, California, HPL-2001-310R1, July 2002
- [8] Tosic V, Pagurek B, Patel K. WSOL-A Language for the Formal Specification of Classes of Service for Web Services [C] // Proc. of ICWS'03 (The 2003 International Conference on Web Services). Las Vegas, USA; CSREA Press, June 23-26, 2003; 375-381
- [9] He Chuan, Gu Lei, Du Bin, et al. A WSLA-based monitoring system for grid service - GSMon [C] // Proc. of 2004 IEEE International Conference Services Computing, 2004. (SCC 2004), Sept. 2004
- [10] WS-Resource Framework. http://www.globus.org/wsrif/specs/ws-wsrf.pdf
- [11] The Apache Software Foundation. "Xerces2 Java Parser", product information, The Apache Software Foundation. http://xerces.apache.org/xerces2-j/xni-design.html (2006-4-5)
- [12] 姚雷,唐亚哲,杨宇彬,等.基于 Xerces2-java 的多 XML 文档关联验证 [J]. 计算机工程与应用, 2007, 45(5): 102-105
- [13] Tomarchio O, Vita L. On the Use of Mobile Code Technology for Monitoring Grid System // Proceedings of First IEEE/ACM International Symposium on Cluster Computing and the Grid. 2001
- [14] UDDI.org. UDDI Technical White Paper (EB/OL). http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, 2000. 9. 6
- [15] 赵铭. 基于服务等级的网格服务发布和发现机制研究 [D]. 硕士学位论文. 西安: 西安交通大学, 2007
- [16] Globus 4 编程指南 [EB/OL]. http://gdp.globus.org/gt4-tutorial/multiplehtml/index.html (2006-4-5)