

基于网格服务的协同虚拟环境

张 亮, 吴慧中, 肖 亮

(南京理工大学计算机科学与技术学院, 南京 210094)

[摘要] 由于计算资源、网络资源以及现有技术的限制,协同虚拟环境的扩展性问题至今没有得到很好的解决。针对这一问题,讨论了大规模协同虚拟环境的通信模式,提出了一个基于开放网格服务结构 OGSA 的协同虚拟环境体系结构,利用网格服务封装将系统功能实现与具体的物理资源相分离以达到系统可扩展的目的。另外,还重点研究了影响协同虚拟环境扩展性的关键技术——虚拟世界管理服务以及相关的支撑服务。最后基于 Globus Toolkit 开发了一个协同虚拟环境示例程序,验证所提出的框架和方法的可行性。

[关键词] 协同虚拟环境; 开放网格服务结构; 虚拟世界管理; 扩展性

[中图分类号] TP391 [文献标识码] A [文章编号] 1009-1742(2007)11-0157-06

1 引言

随着虚拟现实技术特别是分布式虚拟环境技术的发展,人们自然地联想到用虚拟现实技术来构建一个计算机支持的协同工作环境(CSCW, computer support cooperative work),使得参与协作的人能够更加融洽地协作,这种多人共享并协同工作的分布式虚拟环境称为协同虚拟环境(CVE, collaborative virtual environment),又称为协同虚拟现实(CVR, collaborative virtual reality)。CVE的研究开发工作可以追溯到20世纪80年代初,近年来网络技术的飞速发展更进一步地推进了CVE的研究和开发。目前,CVE系统已经在军事仿真、娱乐、教育和医疗等诸多领域得到了应用^[1,2],但是由于CVE系统本身的复杂性和现有支撑技术的限制,构建一个规模可扩展、功能可扩充、高效的异构型CVE系统仍然面临着许多挑战。

通过对CVE应用需求的分析,可以得知限制CVE系统扩展性的关键问题在于CVE所需各类资源(包括硬件资源和软件资源)的不足或异构。近年来,继万维网之后出现了一种面向资源共享的新型网络计算平台——网格计算(grid computing),网格计算的出现为解决影响CVE系统扩展性的资源共

享问题提供了新的思路。

当前网格计算领域的主流体系结构是开放网格服务体系结构OGSA(open grid services architecture)^[3],它结合了Globus标准和面向商业应用的Web服务,把网格计算从科学与工程计算应用扩展到更广泛的以分布式系统服务集成为主要特征的商业应用领域。网格服务是OGSA中的基本概念,它是一种实现了标准接口、行为和约定的Web服务,通过接口提供服务发现、动态服务创建、服务生命周期管理、消息订阅和通知发送等功能。OGSA以服务为中心,把网格中的资源都包装成服务,隐藏各种资源的异构性,用服务这种统一的实体提供共享接口。笔者结合OGSA对CVE系统进行研究,以期利用网格服务来解决CVE系统的扩展性问题。

2 协同虚拟环境通信模式

通信模式是影响CVE扩展性的首要问题^[4~6],目前CVE的通信模式主要有对等模式(peer-to-peer)和客户/服务器模式(client-server)。对等模式允许网上任意两节点直接通信(如NPSNET和MASSIVE等系统),因此消息延迟小,每个节点完全自治,不会因为一个节点崩溃而影响整个系统;但是对等模式的主要问题是扩充性不好,通常节点需要存储每个实

[收稿日期] 2006-07-17;修回日期 2006-08-31

[基金项目] 高等学校博士学科点专项科研基金资助项目(20020288024),长江三角洲重大科技联合攻关资助项目(BE2004400)

[作者简介] 张 亮 (1981-)男,江苏宿迁市人,南京理工大学计算机科学与技术学院博士研究生,主要研究方向为协同虚拟现实、应用层组播

体的表示、处理每个实体的消息、同步每个实体的状态,当插入一个新的节点时,维护整个世界状态的一致性比较困难。客户/服务器模式则要求客户之间的通信必须通过服务器,即客户首先将消息发送到服务器,然后再由服务器将其发送到其他客户(如 SPLINE 和 RING 等系统)。客户/服务器模式将负载从客户端转移到服务器上,便于保持整个世界的一致性;但是这种模式下服务器也容易成为瓶颈,并且客户之间的消息延迟较大。鉴于对等模式和客户/服务器模式都有各自的局限,并且考虑到大规模 CVE 环境中数据通信的多样性,采用混合模式作为 CVE 的通信模式。系统中部署多个服务器,每个服务器负责管理一组客户机,服务器之间采用对等模式进行通信。对于需要集中控制的一致性管理、并发控制、资源管理等功能都配置在服务器上,这时发送方的客户首先将消息传送到服务器,经过处理后,再由服务器转发到各个接收端;客户间不需要集中控制的消息(如实体状态更新消息、客户之间的音频视频数据)则直接通过客户之间的端到端链路进行。

采用多服务器的主要目的是为了共享多台服务器的资源以确保 CVE 系统的扩展性,但多服务器之间的异构资源互联、动态资源分配及安全问题尚缺乏一个有效的分布式计算平台。OGSA 把用通信手段连接起来的资源无缝集成为一个有机的整体,它给用户提供一种基于互联网的新型计算平台,在这个平台上对客户的要求和提供资源的能力之间进行合理的匹配,为用户的请求选择合适的资源服务。可以看出,OGSA 是支持多服务器 CVE 系统的天然框架,因此,采用网格服务对多服务器 CVE 的资源进行封装,由网格服务来解决多服务器之间的异构资源互联、动态资源分配及安全等问题。基于网络的混合通信模式如图 1 所示,客户端程序与部署在多服务器上的网格服务进行交互,不关心网格服务所占用的具体物理资源。

3 基于网络的协同虚拟环境体系结构

结合开放网格服务结构 OGSA,提出了一个基于网格服务的 CVE 体系结构(见图 2),该结构分为 4 个层次:资源层,互联网协议层,网络支撑层,CVE 服务层。

1)资源层 资源层是系统使用到的所有资源的聚合,包括存在于网格环境中的所有计算设备、存储设备、仿真设备、网络资源等硬件资源以及各种文件

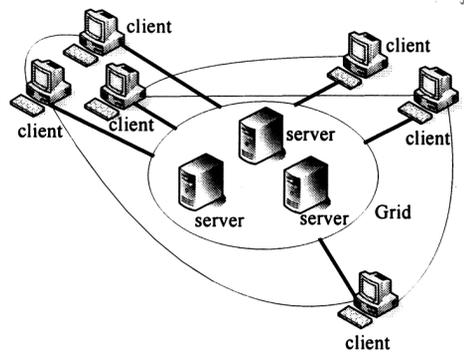


图 1 基于网络的混合通信模式

Fig.1 Mixing communication model based on grid

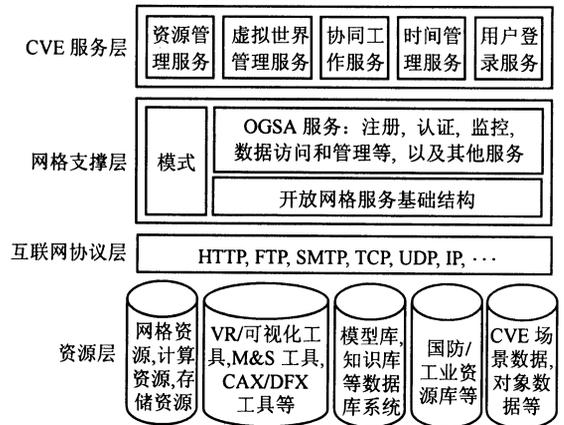


图 2 基于网格服务的 CVE 体系结构

Fig.2 CVE architecture based on grid service

系统和数据库系统等软件资源。CVE 是为具体应用领域服务的,因此除了通用资源外还包括针对于具体应用的资源,如支持 SBA 应用的仿真模型库、知识库、国防/工业资源库、协同设计和仿真工具等。另一方面,CVE 系统本身还包含一些数据资源如场景数据、对象数据、多媒体数据以及用户认证信息和访问控制信息等。

2)互联网协议层 互联网为构建大规模分布式应用提供了天然的网络平台,互联网上存在一系列不同层次的标准协议包括 IP, TCP, UDP, HTTP, FTP 和 SMTP 等,这些协议将分布于各地的资源互联起来,是在互联网上构建分布式应用的通信基础。

3)网络支撑层 网络支撑层包括 OGSA 的主要组件:开放网格服务基础结构、OGSA 服务和 OGSA 模式。这些组件提供了构造网格服务的基本构造块(包括描述和发现服务属性、创建服务实例、管理服务生命期、管理服务组,以及发布和订阅通知的标准

接口和相关行为)和一些相关服务(包括网格信息管理服务、网格资源管理服务、网格数据管理服务、网格作业管理服务以及网格通信和安全服务等)。网格支撑层建立在互联网协议层之上,以互联网协议中的通信、路由等功能为基础,向上为具体应用的网格服务提供网格服务容器环境。

4) CVE 服务层 CVE 服务层建立在网格支撑层之上,是构建可扩展 CVE 系统的关键所在。这一层中将 CVE 系统的服务器端功能封装为网格服务,主要包括:

a. 资源管理服务:在 OGSA 框架下一切 CVE 资源都能封装成网格服务,资源管理服务利用网格服务机制实现资源服务的注册、查找、定位、访问和删除。

b. 虚拟世界管理服务:大规模 CVE 的虚拟世界数据来源主要有场景数据、各种模型数据、多媒体数据、用户认证信息以及访问控制信息,虚拟世界管理服务访问这些数据来构造和维护 CVE 中的虚拟世界。构造和维护 CVE 中虚拟世界是建立可扩展 CVE 系统的关键问题。

c. 协同工作服务:利用全局的知识,对 CVE 中用户协同信息进行管理规划、以便协同完成一项共同的任务。

d. 时间管理服务:提供全局协调机制来保证各客户端推进时间不同的情况下,共享对象能在一致的世界坐标中推进。

e. 用户登录服务:负责管理用户登录和离开虚拟世界,并提供对用户组的管理,这里需要网格信息服务和访问控制等安全服务的支持。

网格服务注册中心用于管理网格服务,记录各服务位置。网格服务注册中心地址是对外公布的,服务结点将网格服务位置注册到注册中心。客户端程序最初并不知道自己所需的服务是由哪些服务结点提供,更不知道服务结点的位置,而是通过查询注册中心获得所需服务位置。注册中心的信息是实时更新的。当服务结点加入网格环境时,注册中心添加其提供的服务;当服务结点退出网格环境时,注册中心删除其提供的服务。注册中心还提供对服务的查询、更新功能,并能处理多个客户端程序的并发请求。

4 协同虚拟环境中虚拟世界管理服务

大规模 CVE 中虚拟世界通常包括复杂的地形

模型、大量的静态实体模型和动态实体模型、逼真的自然景象模型等,导致处理和维持虚拟世界的开销很大。采用虚拟世界空间分区技术对虚拟世界进行管理,将大规模场景分布到多个虚拟世界管理服务(virtual world management service)中,并部署在多个服务器上。

图 3 所示,将虚拟世界 V (构造虚拟世界的数据来自场景数据库 S)用网格分成 N_i 块 regions,记为 R_i ($0 < i < N_i + 1$),然后采用某种策略(例如按照虚拟世界中各 R_i 中的实体数目)将 regions 的集合 SceneManage j ($0 < j < N_i, V = \text{SceneManage}1 \cup \text{SceneManage}2 \cup \dots \cup \text{SceneManage} N_i$)分布到各个虚拟世界管理服务中,一般是每台服务器上部署一个虚拟世界管理服务,从而实现系统负载的分布。当 CVE 场景规模变大时,可以通过添加服务器以及相应的虚拟世界管理服务来扩展系统。

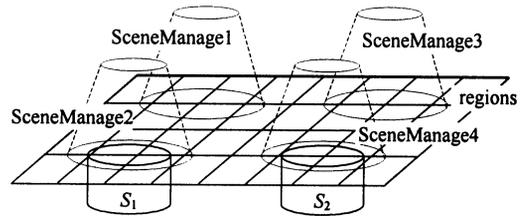


图 3 虚拟世界空间分区示意图

Fig.3 Example of virtual world regions

CVE 系统运行时,动态实体在虚拟世界中的位置和数目会不断变化,在一个时间段,大量的动态实体聚集到一个虚拟世界分区中可能会导致某个虚拟世界管理服务超载。另一方面,在 CVE 中,每个化身都有一个兴趣空间(AOI, area of interest),该兴趣空间受到化身感知能力、环境等因素的限制。当一个化身进入系统时,根据自身的位置信息寻找虚拟世界管理服务,该虚拟世界管理服务结合化身的 AOI 将相应的 region 数据传输给客户端程序。由于化身的 AOI 可能会跨越多个虚拟世界分区,从而引入了多个虚拟世界管理服务之间的协作问题。如果在多个虚拟世界分区交界处附近存在大量的化身,这将使虚拟世界管理服务之间产生大量的通信流量,从而导致某个虚拟世界管理服务超载。因此负载均衡机制采用虚拟世界分区中的动态实体数目、服务器之间的网络通信流量作为各个服务器上的虚拟世界管理服务负载均衡的性能指标^[7]。

利用 OGSA 网格服务可以方便的实现虚拟世界管理服务负载均衡机制。在 OGSA 中,一个特定的

网络服务有一个或多个服务实例,每个服务实例采用服务数据元素(SDE, service data element)来表示自己的状态信息。在此,为虚拟世界管理服务定义两种类型的 SDE: SceneregionType {regionarray: int []}, LoadType {avatarmum: int, networkload: float}。SceneregionType 中属性描述了虚拟世界管理服务实例负责的 regions 标号,为客户端的服务发现提供信息服务。LoadType 中属性描述了场景分区中的动态实体数目、场景分区与其他分区之间的网络通信流量,当属性值超过一定的阈值时,利用 OGSA 的通知机制调用负载均衡服务。在 OGSA 中存在服务工厂(factory)的概念,利用服务工厂可以动态地创建、删除服务实例,维护服务实例生命期。在每台服务器上均部署一个虚拟世界管理服务工厂,CVE 系统初始化时每个工厂以场景分区为初始化参数创建一个分区虚拟世界管理服务。当负载均衡服务检测到某个虚拟世界管理服务超载时,该虚拟世界管理服务负责的一部分 regions 迁移到有多余计算资源和通信资源的虚拟世界管理服务上或者在有多余计算资源和通信资源的服务器上创建新的虚拟世界管理服务,并修改相应的 SDE。

虚拟世界管理服务主要负责虚拟世界的下载与修改,其接口服务以及扩展 WSDL 文件如下所示。

```
public interface Scenemange
{
public void initscene (int regionarray []);
public void downloadscene (String address, int port,
Point position);
public void revisescene (PDU entitypdu);
public socket getreceivepdu_socket ();
...
}
<? xml version="1.0" encoding="UTF-8"? >
...
<gwsdl: portType name = "Virtualworldmanage Port
Type"
extends="ogsi: GridService">
<operation name = "initscene">
<input message = "tns:InitsceneInput Message"/>
<output message = "tns:InitsceneOutputMessage"/>
<fault name="Fault" message = "ogsi: FaultMessage"/>
</operation>
```

```
...
<sd:serviceData name="Sceneregion"
Type="data:SceneregionType"
minOccurs="1"
maxOccurs="1"
mutability="mutable"
modifiable="false"
nillable="false">
</sd:serviceData>
</gwsdl:portType>
```

将抽象接口和具体的编码相分离是 OGSA 的一个重要方面,面向服务系统定义了用于调用服务操作的简单标准机制,采用这种机制的系统可能会在某些环境下遭遇性能问题。虚拟世界管理服务实现既支持多种分布通信协议(如基于 HTTP 的 SOAP 或其他更优的协议),也支持在请求和服务处理发生在同一主机上的交互采用本地通信协议(如本地 IPC)。实现虚拟世界数据下载 downloadscene 接口时,利用 GridFTP 协议;对于虚拟世界数据修改,提供了两个接口, revisescene 直接利用 SOAP 协议将虚拟世界修改 PDU 作为调用参数传输给服务; getreceivepdu_socket 则提供一个接受虚拟世界修改 PDU 的进程 socket,以便直接利用 TCP、UDP 进行数据传输。

虚拟世界管理服务的实现比较复杂,需要和多个功能模块以及服务进行交互,包括兴趣管理服务、一致性控制服务和并发控制服务。

兴趣管理服务(interest management service) 兴趣管理是用来降低网络通信量的一项技术,通过定义对象之间的通信关系和信息的过滤规则,来确定哪些对象可以进行通信以及存在通信关系对象之间的哪些内容需要传输,从而剔除不相关的对象之间不必要的通信以减少整个虚拟环境的网络通信量。这里的兴趣管理服务负责维护各个客户端化身的兴趣空间,为 CVE 虚拟世界管理提供支撑服务。

一致性控制服务(consistency management service) 导致 CVE 环境中时空不一致的原因大致有以下几个方面:各客户端的系统时间不同步,各客户端的空间坐标系不一致,客户端计算和网络传输延时。目前由客户端时间不同步和空间坐标系不一致所带来的时空不一致可以通过时间管理、坐标系变换技术解决。由客户端计算和网络传输延迟所带来的时空不一致却没有被很好的解决,是目前 CVE 领域的一

个研究难点。对于那些事件有因果顺序的情况(例如,分布式仿真坦克的先开火、后击中爆炸),使用服务器作为统一的事件处理机制。将客户端节点发生的事件传输给一致性控制服务,服务对事件排序后,再回传给相关的节点进行处理、显示。对于没有因果顺序的情况(例如,分布式仿真坦克的位置移动),则在客户端采用本地滞后技术或状态推算技术直接将事件传输到相关的客户端节点进行处理、显示,从而减少系统的反应时间。

并发控制服务(concurrent management service)在 CVE 环境中,用户对共享对象的操作可能会发生冲突(例如,在虚拟社区里,用户 A 和用户 B 同时拿起同一个杯子),由于时空不一致(用户 A 已经拿起了杯子,而该事件还没有传输到用户 B),这种冲突就更容易发生。因此,必须要提供并发控制服务来避免冲突的发生,典型的方法有加锁控制和时间戳排序。

图 4 是客户端初始下载虚拟世界数据以及系统运行过程中修改虚拟世界数据的时序图。

5 实例开发

Globus Toolkit 3.0 版提供了一个完整的 OGSi 实现^[8],并且将 Globus Toolkit 以前版本的许多功能重构成与 OGSi 兼容的服务。根据上面提出的框架和技术,将一个 CVE 实例程序——分布式虚拟校园系统(见图 5、图 6)扩展到 Globus3.9.3 平台下。遵循 OGSa 规范,初步实现了虚拟世界管理服务等功能,并在由两台服务器和多台客户机组成的局域网上作了测试,验证了所提出的框架和技术的可行性。值得注意的是 CVE 系统扩展的效率还依赖于实现虚拟世界管理服务时所采用的各种具体算法,需要对这些算法做进一步研究。

6 总结

CVE 系统是一个复杂的系统,由于各类资源以及支撑技术的限制,它的扩展性问题至今没有得到很好的解决。讨论了可扩展 CVE 的通信模式,提出了基于 OGSa 的 CVE 体系结构,利用网格服务封装将系统功能实现与具体的物理资源相分离以达到系统可扩展的目的。另外,还重点研究了 CVE 体系结构中影响协同虚拟环境扩展性的关键问题——虚拟世界管理服务以及相关的支撑服务,为 CVE 系统的扩展性问题提供有效的解决方案。下一步的工作目

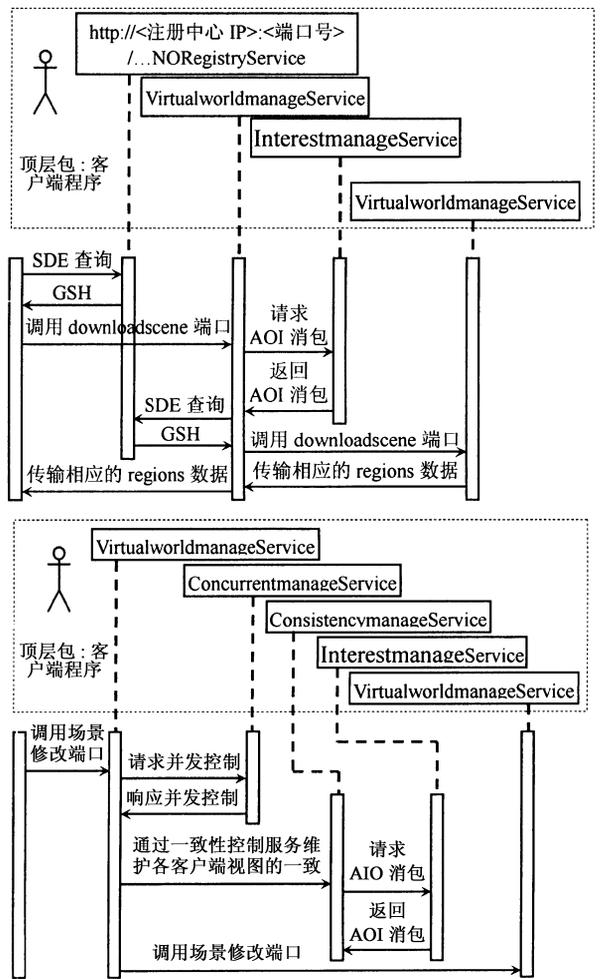


图 4 虚拟世界管理服务时序图

Fig.4 Sequence diagrams of virtual world management service



图 5 分布式虚拟校园系统场景 1

Fig.5 Scene 1 of distributed virtual campus system

标是对兴趣管理技术作进一步研究从而更有效的解决 CVE 系统的扩展性问题。

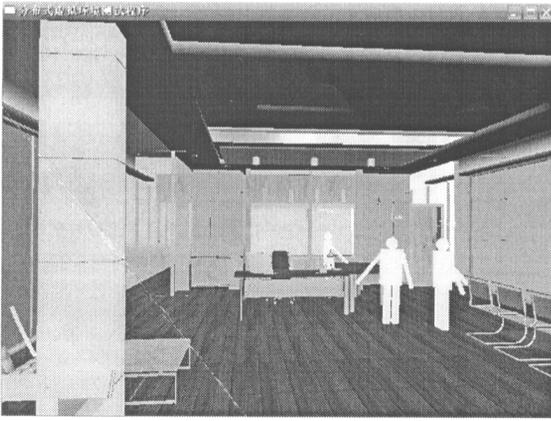


图6 分布式虚拟校园系统场景2

Fig.6 Scene 2 of distributed virtual campus system

参考文献

[1] Greenhalgh C M. Large Scale Collaborative Virtual Environments [D]. The University of Nottingham, 1997

- [2] 潘志庚, 姜晓红, 张明敏, 等. 分布式虚拟环境综述[J]. 软件学报, 2000, 11(4): 461~467
- [3] Foster I, kesselman C. The Grid 2 : Blueprint for a New Computing Infrastructure [M]. Beijing: Publishing House of Electronics Industry, 2004
- [4] 杨孟洲, 姜晓红, 潘志庚, 等. 多用户分布式虚拟环境系统 MUDVE 的实现技术[J]. 计算机辅助设计与图形学学报, 2001, 13(2): 173~178
- [5] Matijasevic M, Gracanin D, Valavanis K, et al. A framework for multiuser distributed virtual environments [J]. IEEE Transactions on Systems, Man, and Cybernetics, 2002, 32(4): 416~428
- [6] Wang Shaoqing, Chen Ling, Chen Gencai. Aframework for Java 3D based collaborative virtual environment [A]. The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings [C]. 2003: 34~39
- [7] Lui J C S, Chan M F. An efficient partition algorithm for distributed virtual environment systems [J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3): 193~210
- [8] Sotomayor B. The Globus Toolkit 3 Programmer's Tutorial [EB/OL]. <http://www.globustoolkit.org>, 2003

Collaborative Virtual Environment Based on Grid Service

Zhang Xu, Wu Huizhong, Xiao Liang

(School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)

[Abstract] Limited by computing resource, network resource and today's technologies, the scalability of collaborative virtual environment (CVE) has not been efficiently solved until now. The paper discusses the communication model of large-scale CVE first and then devises a CVE architecture based on open grid services architecture (OGSA). Utilizing grid service the CVE architecture separates CVE function realizations from physical resource to solve the scalability of CVE. In addition, the paper focuses on the research of virtual world management service and some related services such as interest management service, consistency management service and concurrent management service which are key issues to solve the scalability of CVE. At last, a CVE case based on Globus Toolkit is developed to validate the feasibility of the proposed CVE architecture and methods.

[Key words] collaborative virtual environment; open grid services architecture; virtual world management; scalability