GridSim 剖析

An Anatomy of GridSim

计算网格课程报告
董子龙 10421038
listctrl@163.com
浙江大学 CAD&CG 实验室
2005-6-18

摘要：
计算机集群，网格，点对点网络越来越成为下一代分布并行计算的规范，它们聚集分布的资源解决科学、工程、商业领域的大规模问题。在网格和点对点计算环境中，资源通常通过广域网或因特网连接，在地理上分散于多个管理域，并且由不同的组织以不同的规则拥有和管理。这给资源管理和应用调度的许多领域带来挑战，如安全，资源和规则各异性，纠错，资源条件的多变，政策等。因此，网格计算的资源管理和调度系统不仅要根据资源拥有者和使用者的需求管理资源和应用执行，还要不断地适应资源情况的变化。

在这么大的分布式系统中，资源的管理和应用程序的规划是一项复杂的工程。为了检验资源 Broker 和相关的调度算法的效率，他们的性能需要在各种情况下测试，比如资源的数量，用户的需求。在网格环境中，由于资源和用户是分布在许多组织中，并且有自己的规则，所以几乎不可能可重复地，可控制地测试某个调度算法的性能。于是结果也是无法比较和估计的，GridSim 的出现就是为了解决这个限制。GridSim 是基于 Java 的离散时间网格模拟工具包，支持各异性网格资源，用户和应用模型的建模和模拟。GridSim 提供建立任务和把任务映射到资源，及其管理的接口原语。GridSim 构架分成 5 个层次，报告将剖析每一层的基本内容。

Summary：
Clusters, Grids, and peer-to-peer (P2P) networks have emerged as popular paradigms for next generation parallel and distributed computing. They enable aggregation of distributed resources for solving large scale problems in science, engineering, and commerce. In Grid and P2P computing environments, the resources are usually geographically distributed in multiple administrative domains, managed and owned by different organizations with different policies, and interconnected by wide-area networks or the Internet. This introduces a number of resource management and application scheduling challenges in the domain of security, resource and policy heterogeneity, fault tolerance, continuously changing resource conditions, and politics. The resource management and scheduling systems for Grid computing need to manage resources and application execution depending on either resource consumers' or owners' requirements, and continuously adapt to changes in resource availability.

The management of resources and scheduling of applications in such large-scale distributed systems is a complex undertaking. In order to prove the effectiveness of resource brokers and associated scheduling algorithms, their performance needs to be evaluated under different scenarios such as varying number of resources and users with different requirements. In a Grid environment, it is hard and even impossible to perform scheduler performance evaluation in a repeatable and controllable manner as resources and users are distributed across multiple organizations with their own policies. GridSim is developed to overcome this limitation. GridSim is a Java-based discrete-event Grid simulation toolkit called. The toolkit supports modeling and simulation of heterogeneous Grid resources (both time- and space-shared), users and application models. It provides primitives for creation of application tasks, mapping of tasks to resources, and their management. GridSim system architecture contains five layers; this report will tell the details of each layer one by one.

一、 Introduction

因特网的飞速发展，超强计算机的出现，廉价高速的网络使得人们对大规模分布并行计算提出新的思路，人们希望通过聚集地理上分散开的资源完成大规模计算的任务，这是点对点(P2P)计算和网格(Grid)计算网络的出发点。他们可以对合适的计算资源和数据资源进行发现、共享、选择和聚集。图 1 是网格计算环境的最一般的结构。网格主要分成四层：

1）组织层，包括计算机，网络，科学设备，及其管理系统；

2）核心中间件，包括安全和访问管理，远程作业提交，存储，资源信息等服务，使得安全一致透明的访问远程资源；

3）用户中间件，提供高级工具，如网格资源 Broker，应用开发，自适应运行时环境；
4）应用程序，可以是网格库开发的程序，也可以使用用户中间件将传统的程序转化成网格应用程序。

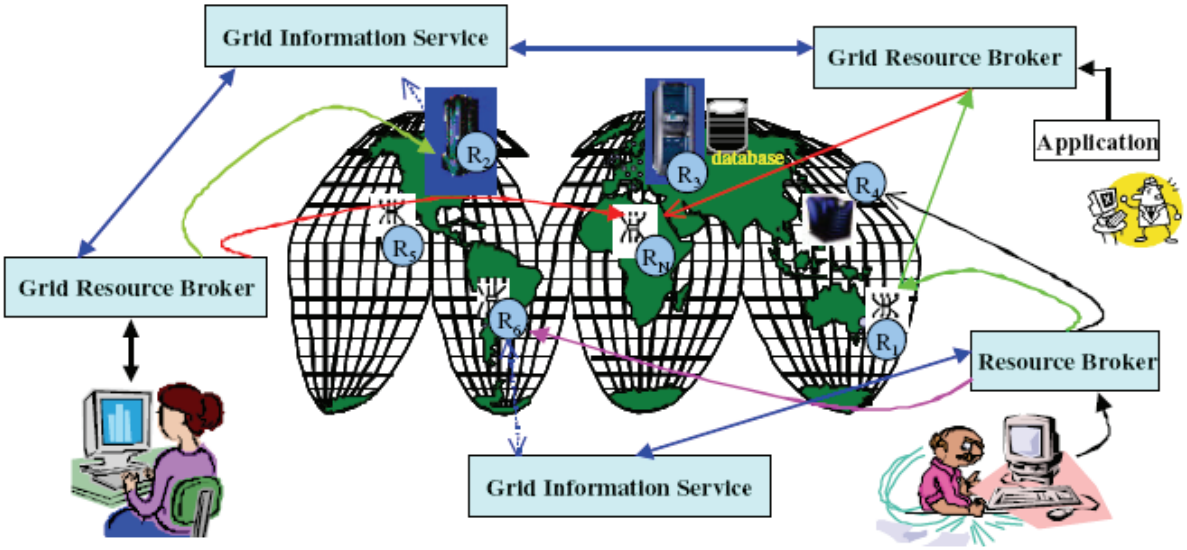

图 1：网格计算环境的一般结构

用户主要和资源 Broker 交互，隐藏了复杂的网格计算结构。Broker 的功能是发现用户可以访问的资源，把作业映射到资源上，准备处理的程序和数据，开始执行作业，最后收集结果。Broker 同时负责监测和跟踪作业完成的进度，自适应于网格运行时环境的变化或资源缺失。

网格计算环境中充满了各异性的分散资源，在这样一个复杂的环境中，资源管理不能采用传统的集中式方法，因为不可能定义系统级别的性能指标和组织访问规则。其它方法可以管理分布式资源：层次调度，分散调度，或两者的结合。人类经济体同样有复杂的各异性和分散性，所以经济可以作为网格计算环境的一个比较，于是可以把经济中市场驱动的一些评判规则和运作方法作用于网格。

网格资源管理和调度策略的研究者们需要一个稳定的框架来模拟资源和应用程序，以评估他们的算法。因为他们大部分人都没有可以利用的测试环境，而即使有这样的网格资源，也是有限的。由于网格的高度复杂性，算法的运行在实际中不可跟踪，也就不能有效地获得算法通用性、自适应性和性能等信息。GridSim 的提出就是为了提供给研究者这么一个可重复、易跟踪的模拟网格。

GridSim 工具包支持大量不同资源的建模和模拟，比如单处理器或多处理器，共享的分布式存储计算机，工作站，SMP，不同能力和配置的计算机集群等。它可以用来模拟多种类型的分布并行计算系统的应用程序调度，如集群，网格，点对点网络。网格和点对点系统中的调度器叫做资源 Broker，Broker 直接面向用户需求，集中增强某个应用程序的性能。

GridSim 可以模拟不同能力，配置和域的网络状况，支持应用程序组合，资源发现信息服务，将作业映射到资源并且控制他们的运行等。这些特色可以模拟资源 Broker 的运行，评估调度算法的性能。

二、 System Architecture

GridSim 工具包提供了全面的工具模拟不同种类的各异性资源，用户，资源 Broker，应用程序，和调度器。它的特色在于：

1) 可以模拟不同类型的资源；
2) 资源的操作模式包括 Timer-shared 和 Space-shared；
3) 可以定义资源的能力，以 MIPS 或 SPEC 为标准；
4) 资源可以定位于不同的时区；
5) 根据资源的当地周末和假期模拟非网格负载；
6) 可以预订资源；
7) 模拟不同并行模型的应用程序；
8) 程序也可以是各异性的，CPU 密集型或 I/O 密集型；
9) 一个资源上提交的作业数没有限制；
10) 多用户可以同时提交任务给一个资源；
11) 定义资源之间的网络速度；
12) 支持模拟动态和静态调度；
13) 可以记录所有或者选定的操作的统计数据，并用 GridSim 数据分析方法进行分析。

GridSim 的架构是层次化和模块化的，可以将现有的技术作为独立的元件纳入其中。图 2 是 GridSim 平台及应用的多层次构架抽象。

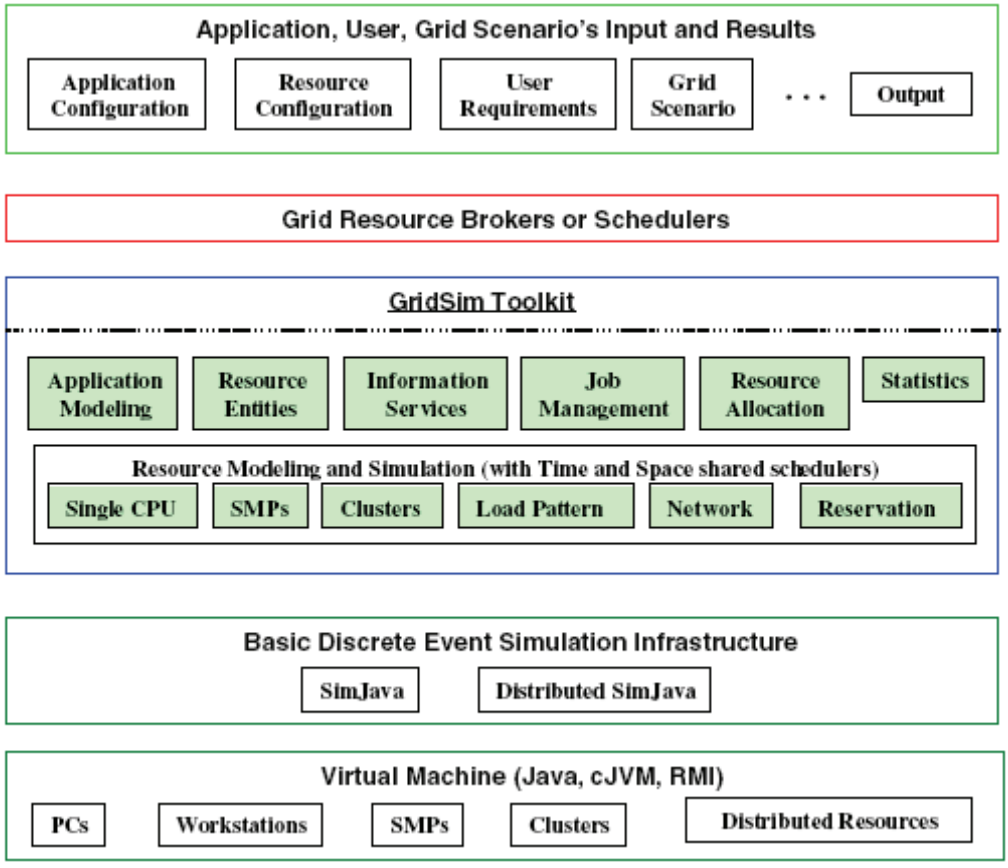

图 2：GridSim 平台和元件的模块化构架

第一层，Java 接口和 Java 虚拟机（JVM），JVM 支持单个或多处理器，包括集群。

第二层，利用第一层提供的接口，建立基本的离散事件基础结构，比较流行的是SimJava。

第三层，包括核心网格实体模拟——资源，信息服务等，应用程序模型，统一访问接口，应用模拟原语，和建立更高层次实体的框架。主要利用第二层的离散事件服务模拟系统实体。

第四层，模拟资源聚集，网格 Resource Broker 或 Scheduler。

第五层，不同情境下的资源和应用建模，利用第三层和第四层提供的服务，评估调度，资源管理策略，算法。

三、 SimJava——A Discrete-Event Infrastructure Package

SimJava 的设计目标是基于 C++的 ASE++库，建立一个纯文本的 Java 离散事件模拟库。模拟由一个中央类 Sim_system 控制，系统中包括一个 Sim_entity 对象的列表，和一个 Evqueue 对象 *future* 保存将来事件，还有一个 Evqueue 对象 *deferred* 保存延迟事件。

3.1 Sim_system

Sim_system 是完全静态的类，不应有任何实例化的对象，SimJava 的基本操作最后大多归结到 Sim_system 的静态成员函数。初始化完成后，包括所有的 Sim_entity 都已经建立并加到 Sim_system 中，Sim_entity 的 Sim_port 之间的连接也设置好，Sim_system 就进入 run()函数的主循环中。

```
static public void run() {
    System.out.println("Sim_system: Entering main loop");
    run_start();
    while(run_tick()) { }
    run_stop();
  }
```

run_start()，启动所有 Sim_entity 的线程。

run_tick()，让所有处于 RUNNABLE 状态的 Sim_entity 运行一个 tick，然后，处理 *future* 队列的第一个事件，以及时间戳与这个事件相同的时间。如果队列为空，就可以让 Sim_system 停止循环了。

run_stop()，停止所有的 Sim_entity。

3.2 Sim_entity

所有的用户模拟对象都从这个类继承，通过重写 body()函数实现实体的具体操作。body()在内部函数 run()中调用，run 会在线程创建后自动执行。

Sim_entity 基本的模拟操作包括：

Sim_schedule，发送时间对象给其他实体，根据不同需要有 5 个重载函数，最终都是通过 Sim_system 的静态成员 send()把事件对象加入 *future* 队列；

Sim_hold，挂起实体对象，通过 Sim_system 的静态成员 hold()把 HOLD_DONE 事件加入 future 队列，以在一定时间后重新运行，然后停止当前实体；

Sim_wait，停止等待一个事件，通过 Sim_system 的静态成员 wait()，将当前实体的状态设为 WAITING，反复地暂停重始，检测事件缓冲是不是有新的事件。

通过以上的 3 个操作，可以通过发送和接收时间，高效地构造一个动态实体的网络。

四、 Simulation of Core Grid Entities

模拟过程中，GridSim 创建大量多线程实体，每一个都运行在自己的线程中，实体的行为在 body()方法中模拟。基于 GridSim 的模拟包括 user，broker，resource，information service，statistics，和 network based I/O，如图 3。

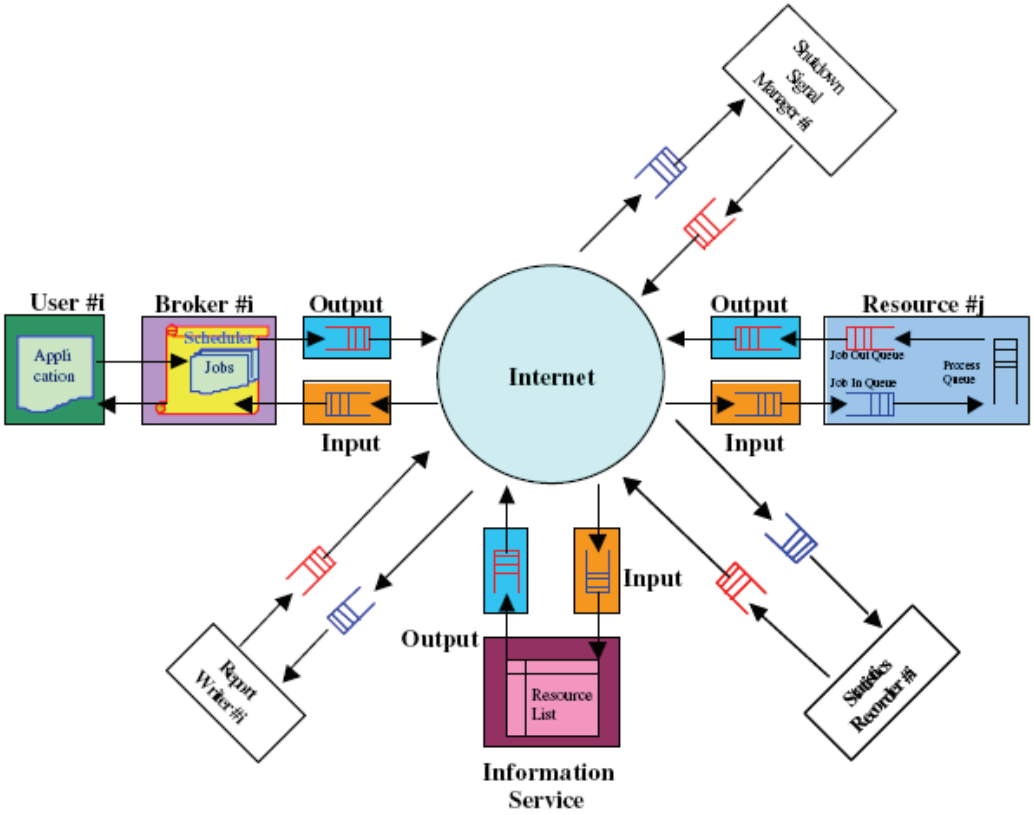

图 3：GridSim 模拟流程

4.1 User

每一个 User 实体实例对应一个网格用户，有以下性质：

1） 作业的类型；
2） 调度优化策略；
3） 活跃度；
4） 时区；
5） 截止时间和预算；
6） D 和 B 因子。

4.2 Broker

每一个用户连接一个 Broker 实例。用户的所有作业都提交给 Broker，然后根据用户的原则调度任务。Broker 首先要从全局目录中动态获得可用资源的列表，所有的 Broker 都是从自己的用户的利益出发，必须竞争有限的资源。所以，Broker 的调度算法要高度适应市场的供需状况。

4.3 Resource

每一个 Resource 实体实例对应一个网格资源，有以下性质：

1） 处理器个数；

2） 运行成本；

3） 运行速度；

4） 内部调度原则，time-shared 还是 space-shared；

5） 当地负载因子；

6） 时区；

只要从网格信息服务获得资源的连接细节，Broker 就可以直接查询资源的属性。

4.4 Grid Information Service

提供资源注册服务，保存网格可用资源的列表，供 Broker 查询。

4.5 Input and Output

GridSim 实体之间的信息传输通过 Input 和 Output 实体，他们的 I/O 通道或端口就是到 Input 和 Output 实体的连接。独立的 Input 和 Output 实体可以摸拟双工通信和多用户并行通信，也可以透明地模拟通信延迟。

五、 Modeling of Application and Resource

5.1 Application Model

GridSim 没有明确定义应用程序模型，用户可以自由选择。每一个任务可能有不同的处理时间和输入输出大小。GridSim 的 Gridlet 对象定义这些参数，包括用户的需求。

Gridlet 包涵任务的所有信息，这些基本的参数用来决定执行时间，和在用户和远程资源之间传输输入输出文件所需要的时间。Gridlet 也作为结果的返回类。

5.2 Resource Model

GridSim 可以创建各种速度的处理单元（PE），一个或多个处理单元构成机器，一台或多台机器可作为一个网格资源。因此，网格资源可以是单处理器，SMP，或计算机集群。单 PE 或 SMP 网格资源通常运行 time-shared 操作系统，多任务调度使用 round-robin 策略。而分布式存储多处理系统运行队列系统，使用 space-shared 调度，即在特定的 PE 上执行 Gridlet。调度策略可以是 first-come-first-served，back filling，shortest-job-first-served 等。高端的 SMP 也可以使用 space-shared 调度。

由于现实系统中的任务调度非常复杂，GridSim 的调度是面向进程的离散事件中断过程，时间间隔就是完成需时最短的作业的时间。GridSim 通过发送、接收、调度事件来模拟的作业执行。根据调度策略和队列或执行中的作业个数，GridSim 调度发给自己的事件模拟资源分配。

5.2.1 Scheduling in Time-shared Resource

GridSim 资源模拟器利用内部事件模拟 Gridlet 的执行和 PE 资源的分配。当一个新的 Gridlet 作业到达资源，GridSim 更新已存在的 Gridlet 的处理时间，并把新的作业加入执行

集中。每一次重新调度，都要在执行集中最短作业完成时，计划发送一个内部事件。然后等待这个事件。

5.2.2 Scheduling in Space-shared Resource

Space-shared 资源也是利用内部事件模拟调度。当你一个新的作业到达资源，如果有空闲 PE，系统立即开始执行这个作业，否则，加入等待队列。在 Gridlet 分配过程中，决定作业处理时间，并在执行时间结束时计划发送一个事件。当 Gridlet 作业完成，立刻发送一个内部事件表明作业结束。资源模拟器接收到这个事件，释放完成的 Gridlet 占用的 PE，并检查等待队列中是否有其它的作业。如果队列中有作业，根据预定的策略选择合适的作业分配到空闲的 PE 上。

六、 Simulation of Grid Resource Broker

为了模拟网格 Resource Broker，研究者需要创建网格用户和调度系统的新实体。用户定义的实体从 GridSim 的并行实体继承，具有通过事件通信的能力。具体的步骤设计资源和应用程序的模拟，Broker 的模拟。

6.1 模拟网格环境的步骤

模拟一个可用于分析调度算法的网格环境有四个步骤：

1） 创建各种能力和配置的网格资源，和不同需求的用户；
2） 创建大量的 Gridlet 为应用程序建模，同时定义任务的参数；
3） 创建与我们的 Resource Broker 交互的用户实体；
4） 最后，我们要实现一个 Resource Broker 完成应用程序的调度，和资源分配。

6.2 Broker 的基本构架

图4 是Resource Broker的基本构架和他与其他实体之间的交互流程。Broker的主要元件是实验接口，资源发现和交易，调度流程管理，Gridlet分发器，Gridlet接收器。元件的功能及相互之间的通信如下：

1） 用户实体创建实验实例，然后通过实验接口，把描述应用程序的 Gridlet 列表和用户需求传递给 Broker；
2） Broker 资源发现和交易模块从 GridSim GIS 实体获得资源的联系信息，然后与资源交互建立配置和访问代价；Broker 有一个 Broker Resource 列表维护资源属性，每一项资源有一个提交执行的 Gridlet 列表，资源的性能数据通过测量外推法预测；
3） 调度流程管理根据用户的需求选择合适的调度算法把 Gridlet 映射到资源，Gridle 将被加入到相应的 Broker Resource 的 Gridlet 列表中；
4） 对于每一个资源，Gridlet 分发器会根据资源的使用策略选择执行的 Gridlet 数量，避免超过资源的负载；
5） Gridlet 分发器使用 GridSim 异步服务，把 Gridlet 提交给资源；
6） Gridlet 处理完成后，资源把它返回给 Broker 的 Gridlet 接收模块，由接收模块测量和更新运行时参数，帮助预测作业完成率和决定调度；

重复3-6的步骤，直到所有的Gridlet都处理完，或者Broker过了极致时间或预算限制。最后，Broker返回实验数据和处理过的Gridlet给用户实体。

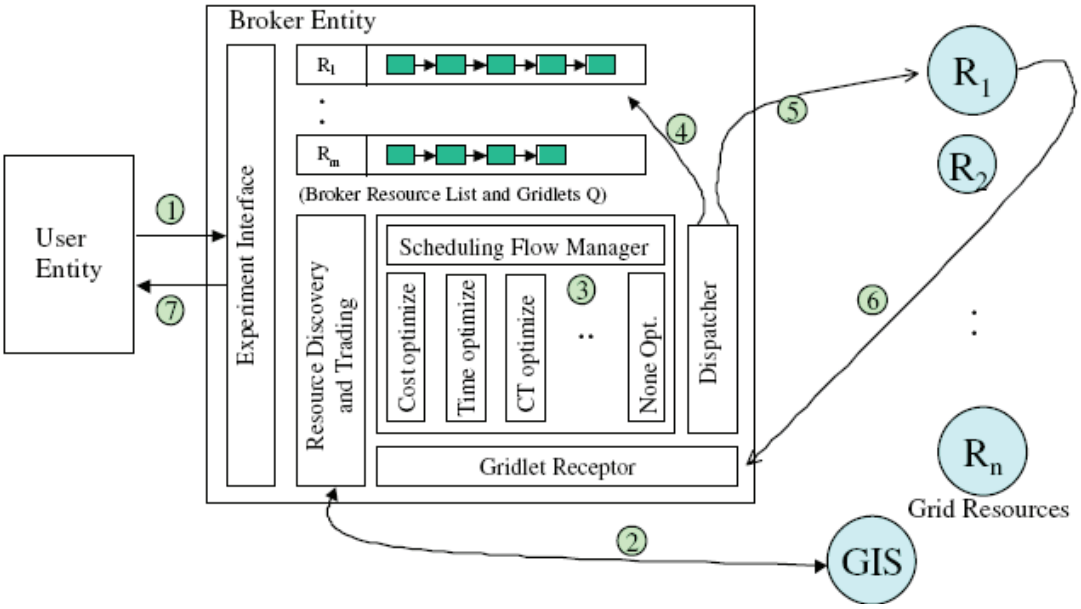

图4：网格Resource Broker系统构架及其他实体的交互

七、 Conclusion

GridSim 是一个面向对象的资源建模和调度模拟工具。GridSim 能模拟不同能力，时区，配置的 Time-shared 和 Space-Shared 资源；支持不同的应用程序模型，通过模拟的应用调度器映射到资源执行。以上介绍了 GridSim 的构架和元件，包括创建基于 GridSim 的应用程序调度模拟器的步骤。

八、 Reference

Anthony Sulistio and Rajkumar Buyya, A Grid Simulation Infrastructure Supporting Advance Reservation, Proceedings of the 16th International Conference on Parallel and Distributed Computing and Systems (PDCS 2004), November 9-11, 2004, MIT, Cambridge, USA, pp. 1-7.

Fred Howell and Ross McNab (1998) Simjava: a discrete event simulation package for Java with applications in computer systems modelling, in proc. First International Conference on Web-based Modeling and Simulation, San Diego CA, Society for Computer Simulation, Jan 1998.

Rajkumar Buyya and Manzur Murshed, GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Volume 14, Issue 13-15, Wiley Press, Nov.-Dec., 2002.

Manzur Murshed and Rajkumar Buyya, Using the GridSim Toolkit for Enabling Grid Computing Education, International Conference on Communication Networks and Distributed Systems Modeling and Simulation (CNDS 2002), January 27-31, 2002, San Antonio, Texas, USA.