

GridSim 仿真代码自动生成器 GridsimHelper

邓 蓉 陈 闯 中 李 灿 王小明 李 捷 张军旗

(同济大学计算科学与技术系 上海 201804)

摘 要 基于离散事件仿真平台 simJava 的网格建模和仿真工具 GridSim 提供了大量用于仿真分布式系统的基础类。但是使用该工具进行仿真并不简单。除了编制大量代码之外,初学者还需要花费较长时间熟悉实验构造过程和各类基础类的使用方法。针对此问题,设计、实现了图形界面(Graphical User Interface, GUI)的仿真代码生成器 GridSimHelper,旨在缩短 GridSim 平台的学习过程,减轻用户实施仿真实验的工作量。该工具允许用户对自行设计的调度算法进行仿真测试,并提供静态启发式算法 min-min 和 max-min 作为性能比较的对象。详细介绍了 GridSimHelper 的设计思路和实现细节。仿真实验结果证实了该工具的正确性和有效性。

关键词 网格,仿真平台,GridSim,代码生成器

GridsimHelper: A Code Generator for GridSim Platform

DENG Rong CHEN Hong-zhong LI Can WANG Xiao-ming LI Jie ZHANG Jun-qi

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

Abstract The Grid Modeling and Simulation toolkit GridSim provide a lot of basic tools to simulate essential components of distributed system. However, the process of simulation is not simple. Except for the necessity to write Java code that used the GridSim toolkit packages; new learner must experience the long learning curve in order to utilize the toolkit functionalities for effective simulations. For this end, we designed and implemented a GUI based code generator for GridSim, which is called GridsimHelper, with the aim of shortening the learning curve and enable fast creation of simulation models. The code generator provides facility for users to input and evaluate the performance of any scheduling algorithm, as well as baseline algorithm (provided by GridsimHelper) min-min and max-min for comparison. This paper gave the implementation details of GridsimHelper, and the experimental results illustrate the correctness and usefulness of GridsimHelper.

Keywords Grid, Simulation platform, GridSim, Code generator

1 引言

2002 年,开源的网格仿真器 GridSim^[1]正式对外公布,与目前常用的其它仿真器(Bricks, MicroGrid 和 SimGrid)相比^[2],其主要优势在于它不仅提供了网格^[3]的各种基本功能部件,而且仿真了所有部件的基本行为。熟悉 GridSim 平台的程序员能够不受约束地设计和测试调度算法的性能,甚至更改基本功能部件,对网格体系结构进行仿真。灵活的使用方式和广阔的应用前景使 GridSim 自诞生以来就一直备受关注。在我国,仅 2006 年至今就有 56 篇以上的文献使用该平台进行仿真实验^[4,5]。

但是,使用 GridSim 工具包并不轻松。为了进行有效的仿真,使用者必须经历一段漫长的学习过程,掌握 GridSim 工具的使用方法。每验证一种策略、算法或是模拟一种网格体系结构,使用者必须编写完整的 Java 程序搭建实验环境,将

仿真所涉及的所有对象有机地组合起来。不同仿真程序的编制过程存在许多重复劳动,大大降低了仿真实验的效率。列举一例:GridSim 提供了一个名为 Example6.java 的用户代理示例程序,目的在于演示 GridSim 仿真环境的配置和使用细节。该程序长 436 行,调用 GridSim 平台 65 次以上。除语句 id = GridSim.rand.intSample(this, totalResource_)随机为任务分配处理器之外,其它所有语句均用于和 GridSim 平台进行交互。在所有对调度算法进行性能评估的 GridSim 仿真程序中,这部分代码结构非常相似。

为了最大限度地减轻仿真实验的构造难度,我们开发了 GridsimHelper。该工具为仿真实验提供图形用户接口,通过接收资源、任务属性和应用程序调度算法,自动生成仿真代码。A. Sulistio 的 Gridsim 代码生成器 Visual Modeler (VM)^[6]与本文类似,但 VM 没有为用户提供编辑调度算法的窗口。为了产生用于性能评估的 GridSim 仿真程序,算法

到稿日期:2009-11-04 返修日期:2010-01-26 本文受 863 国家重点基金项目(No. 2009AA01Z141),国家青年自然科学基金(No. 60803065),中国博士后特别资助基金(200801029)资助。

邓 蓉(1973—),女,讲师,主要研究方向为操作系统、分布式计算等,E-mail: drong2003@gmail.com;陈闯中(1951—),男,博士生导师,主要研究方向为操作系统、分布式计算等;李 灿(1985—),男,硕士,主要研究方向为分布式计算;王小明(1986—),男,硕士,主要研究方向为分布式计算;李 捷(1979—),男,博士生,主要研究方向为分布式计算;张军旗(1979—),男,博士后,主要研究方向为分布式计算、群智能系统。

的设计者必须重新对 VM 生成的代码进行编辑,加入自己编写的调度算法。本文提出的 GridsimHelper 更适合初学者或不熟悉 Java 程序设计的人员使用;可以为任何针对独立任务的调度算法生成适合在 GridSim 平台上运行的仿真程序。仿真实验涉及的调度算法可以是 GridsimHelper 提供的基准算法 min-min 或 max-min,也可以是用户通过 schedule algorithm 选项卡中的文本框输入的其它调度算法(见图 3)。使用 GridsimHelper 工具,不熟悉 GridSim 的用户也可以进行仿真实验,从而大大减小了仿真实验的工作量。

本文第 2 节简介 GridsimHelper 工具的功能及其与 GridSim 仿真平台之间的关系;第 3 节是本文的重点,详细介绍了 GridsimHelper 的设计和实现细节;第 4 节介绍仿真实例。

2 GridsimHelper 的功能

GridsimHelper 工具为用户自动生成可以运行在 GridSim 平台的 Java 代码。图 1 为 GridsimHelper 工具与 GridSim 平台之间的关系:GridsimHelper 的使用者通过图形用户接口输入仿真实验的环境参数、被调度任务集合的所有属性,并选择或输入调度算法(步骤 1)。之后,GridsimHelper 根据输入自动生成仿真代码 Experiment.java(步骤 2)。初学者可以直接在 GridSim 仿真平台上运行该程序(步骤 3)。

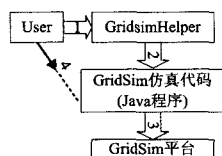


图 1 GridsimHelper 与 GridSim 仿真平台之间的关系

GridsimHelper 生成的所有仿真代码(重命名后)均有保存价值,可以在 GridSim 平台上重复运行。多次使用后,GridsimHelper 的用户将拥有一个仿真实验库。稍熟练的用户可以直接对库中的仿真程序进行编辑,构造新的仿真试验(步骤 4)。例如,用户可以保留原仿真代码中所有任务的属性,修改网格环境参数(资源数量、属性等)或者使用新的算法对集合中的任务进行调度。

目前,GridsimHelper 的功能仅定位于自动生成用于验证调度算法的 GridSim 仿真程序。原因是验证调度算法时,仿真实验无需改动 GridSim 基本功能元件之间的交互过程,被测试的调度算法仅与用户对象的行为相关,且仅涉及其中处理器分配的一小段代码;故 GridsimHelper 可以为用户自动生成除调度算法之外的所有仿真代码,仅以待调度任务列表和资源列表为接口调用用户自己编写的调度算法,并接受其返回的调度方案。

GridSim 平台虽可用于对网格体系结构进行性能评估,但实验难度巨大,远非初学者可为。这是因为,此类仿真实验需要修改几乎所有网格元件的基本行为,代码自动生成工具与用户之间的接口过于复杂。故至少在本阶段,我们将 GridsimHelper 的功能定位于仅为独立任务调度算法自动生成仿真代码。

3 GridsimHelper 的设计与实现

GridsimHelper 工具主要包含 6 个 java 程序(Resource-

Machine.java, ResourceProperty.java, GridletProperty.java, User-Property.java, AlgGenerator.java, CodeGenerator.java) 和 1 个仿真程序模板 Model.txt,见图 2。

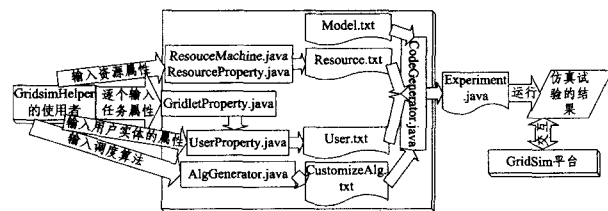


图 2 GridsimHelper 的结构

GridsimHelper 构造的仿真代码名为 Experiment.java,由 4 个文本文件拼接而成。它们是 Model.txt,Resource.txt,User.txt 和 customizeAlg.txt。其中,

- Model.txt;包含 Experiment.main(),User.body()和其它所有固定的仿真代码段,是仿真程序的核心框架。该段代码与网格配置参数和调度算法无关,适用于所有实验。

- Resource.txt;ResourceMachine.java 和 ResourceProperty.java 生成所有 GridResource 对象。具体而言,使用用户从 GUI 输入的资源属性,自动生成包含 Experiment.create-resource()方法的文本文件 Resource.txt。用户输入的网格资源属性包括资源数量、调度算法(时分、空分或其它)、资源的波特率(资源与外界网络链接的传输速度)、高峰(低谷)时的本地负载(忙百分比),以及每个资源所包含的处理器个数和每个处理器的处理能力等。

- User.txt;User.txt 文件存放系统(UserProperty.java 和 GridletProperty.java)根据用户输入的任务属性自动生成 Experiment.createUser()方法;该方法主要用于创建仿真系统中的所有 User 对象。若实验的目的是验证静态调度算法的效率,Experiment.createUser()方法仅创建一个 User 对象,与 GridResource 和 GIS 交互,独享网格中的所有资源。当仿真系统中存在多个 User 对象时,每个 User 作为独立的行为实体与其它 User 竞争网格资源,完全工作在动态、逼真的网格环境中。具体而言,UserProperty.java 使用包括用户名、波特率等,除调度算法之外 User 的所有系统属性来创建 User 实体。GridletProperty.java 为 User 实体所代理的所有任务生成 Gridlet(GridSim 将任务建模为 Gridlet)。

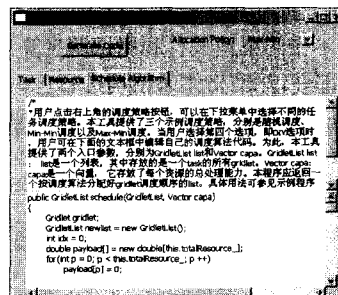


图 3 Schedule Algorithm 选项卡

- customizeAlg.txt;图 3 中的文本框供用户编辑自己的调度算法。例如在其中输入遗传算法代码后,algGenerator.java 生成遗传算法调度实验所需的 schedule()方法,将其存入文件 customizeAlg.txt。为了指导用户编写正确的调度算法,GridsimHelper 提供了 3 个示例调度策略,分别是随机调度、min-min 调度以及 max-min 调度。

按下 Generate Code 按钮后,GridsimHelper 将拼接文本 Model.txt,Resource.txt,User.txt 和 customizeAlg.txt,形成完整的 Gridsim 仿真代码 Experiment.java。

4 实验与分析

为了证明系统的正确性,我们首先利用 GridsimHelper 创建了一个含 5 个资源的异构网格,令其使用 min-min 调度算法运行 45 个独立的任务。图 4 是被仿真的网格、等待调度的任务以及仿真实验的结果片断。实验表明,GridsimHelper 自动生成仿真代码完全可以在 GridSim 平台上运行。

资源名称	资源 ID	资源类型	资源能力	资源位置	资源状态
Resource 1	1	Intel Pentium 4	1000	Time-shared	3.0
Resource 2	2	Compaq AlphaServer	2000	Time-shared	8.0
Resource 3	3	Intel Pentium 4	2175	Time-shared	0.0
Resource 4	4	Intel Pentium 4	2024	Time-shared	0.0
Resource 5	5	IBM Ultra 5	1934	Time-shared	7.0

任务 ID	任务名称	任务长度 (MI)	任务开始时间	任务结束时间	任务完成时间
0	Task 0	274.0	1740.40	1826.25	45.85
1	Task 1	274.0	1740.40	1826.25	61.62
2	Task 2	274.0	1740.40	1826.25	41.61

图 4 GridsimHelper 产生的代码在 Gridsim 上的运行结果

下一组实验针对 GridsimHelper 为 max-min, min-min 算法自动生成仿真代码。算法 max-min 和 min-min 为静态调度算法,故每次实验只有 1 个用户。3 次实验分别用来调度 50, 100 和 150 个任务。考察指标有二:调度长度(makespan)和负载平衡度。表 1 和表 2 为仿真实验的结果,从中得出结论: min-min 算法具有较小的 makespan,但资源负载平衡度较差(不同的资源,完成系统指派给它的所有任务所消耗的时间相差很大)。与此相反,虽然 max-min 在指标 makespan 上略逊于前者,但负载平衡度明显较优。综上所述,算法 min-min 和 max-min 各有千秋。此结论与文献[7]所述相符,系统提供的调度算法正确无误。

结束语 为了减少 GridSim 仿真代码的编写难度,使不熟悉 Gridsim 环境和 Java 编程的调度算法设计者也能够轻松地编写出可以在 GridSim 平台上运行的仿真实验代码,本文具体地提出并实现了一个具有图形用户接口的 GridSim 代码生成器 GridsimHelper。它可以基于实验者输入的网格资源特性和需要运行的任务特性构造实验环境,并允许用户随意

选择和编写调度算法。实验结果表明,GridsimHelper 的代码生成过程以及系统所提供的两种经典调度算法正确,确实达到了减轻调度算法设计者工作强度的目的。

表 1 GridsimHelper 为算法 min-min 生成的仿真代码的性能

任务数	任务长度(MI)		单个资源完成所有任务的时刻		资源的负载不平衡程度 (s)
	最小	最大	最早	最晚	
50	274.0	9706.0	1586.54	1795.67	209.13
100	274.0	9706.0	3149.48	3366.18	216.70
150	274.0	9706.0	4810.63	4989.62	178.99

表 2 GridsimHelper 为算法 max-min 生成的仿真代码的性能

任务数	任务长度(MI)		单个资源完成所有任务的时刻		资源的负载不平衡程度 (s)
	最小	最大	最早	最晚	
50	274.0	9706.0	1780.40	1826.25	45.85
100	274.0	9706.0	3345.73	3407.35	61.62
150	274.0	9706.0	4954.96	4996.57	41.61

参考文献

- [1] Buyya R, Murshed M. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing [J]. The Journal of Concurrency and Computation: Practice and Experience (CCPE), 2002, 14 (13-15): 1175-1220
- [2] 卢鹏, 金海, 谢夏, 等. 关于网格模拟器的研究[J]. 高性能计算技术, 2005, 173(2): 5-9
- [3] Foster I, Kesselman C. The Grid: Blueprint for a new computing infrastructure[M]. 1th ed. San Francisco: Morgan Kaufmann Publishers, 1998: 1-20
- [4] 刘宴兵, 杨茜慧, 王文斌. 基于 GridSim ToolKits 的网格仿真环境设计与实现[J]. 计算机科学, 2008, 35(6): 83-85
- [5] 李炯, 卢显良, 董仕. 基于 GridSim 模拟器的网格资源调度算法研究[J]. 计算机科学, 2008, 35(8): 95-97
- [6] Sulistio A, Chee Shin Yeo, Buyya R. Visual Modeler for Grid Modelling and Simulation (GridSim) Toolkit[C]//Proc. of the 3rd International Conference on Computational Science (ICCS 2003, LNCS Series). Melbourne: Springer Verlag Publications, 2003: 1123-1132
- [7] Braun T D, Siegel H J, Beck N. A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems[J]. Journal of Parallel and Distributed Computing, 2001, 61(6): 810-837

(上接第 117 页)

- [2] 张燕, 傅建明, 孙晓梅. 一种基于模型检查的入侵检测方法[J]. 武汉大学学报: 理学版, 2005, 51(3): 319-322
- [3] Duan Z, Tian C, Zhang L. A decision procedure for propositional projection temporal logic with infinite models[J]. Acta Informatica, 2008, 45(1): 43-78
- [4] Rohrmair G T, Lowe G. Using data-independence in the analysis of intrusion detection systems[J]. Theoretical Computer Science, 2005, 340(1): 82-101
- [5] Sheyner O, Haines J, Jha S, et al. Automated generation and analysis of attack graphs[C]//Proceedings of the 2002 IEEE Symposium on Security and Privacy. 2004: 273-284
- [6] Nan Z, Mark R, Dimitar P G. Evaluating Access Control Policies

Through Model Checking[C]//Lecture Notes in Computer Science, Information Security. Springer Berlin, 2005, 3650: 446-460

- [7] Sebastian S, Michael V, Hartmut K. Identifying Modeling Errors in Signatures by Model Checking[C]//Lecture Notes in Computer Science, Model Checking Software. Springer Berlin, 2009, 5578: 205-222
- [8] Olivain J, Goubault-Larrecq J. The Orchids Intrusion Detection Tool[C]//Proceedings of the 17th International Conference on Computer Aided Verification. Lecture Notes in Computer Science. Springer, Edinburgh, Scotland, UK, 2005, 3576: 286-290
- [9] 朱维军, 王迺冉, 周清雷. 一种基于投影时序逻辑模型检测的入侵检测方法[J]. 网络安全技术与应用, 2010, 3: 25-27