

基于 GridSim 的网格模拟框架设计与实现

胡志刚, 李林

(中南大学信息科学与工程学院, 长沙 410083)

摘要: 设计并实现一个基于 GridSim 的网格模拟框架 GSF, 利用 XML 语言描述网格资源、用户、作业, 提供网格调度接口。针对 workflow 作业定义 workflow 描述语言 GSWDL, 实现一个 workflow 模拟器 WorkFlowEngine。模拟实验结果证明, GSF 可以减少用户对 GridSim 的学习时间和难度, 为研究者提供一个易用、可扩展的网格模拟环境。

关键词: 网格模拟器; 模拟框架; 工作流; 可扩展标记语言

Design and Implementation of Grid Simulation Framework Based on GridSim

HU Zhi-gang, LI Lin

(School of Information Science and Engineering, Central South University, Changsha 410083)

Abstract This paper designs and implements GridSim Simulation Framework(GSF), which describes grid resource, user and job by using XML, and provides an interface for grid scheduling. Aiming at workflow jobs, it defines workflow description language named GridSim Workflow Description Language(GSWDL) and provides a workflow simulator called WorkFlowEngine. Simulation experimental result shows that GSF provides an easy-to-use and extensible grid simulation environment for researchers, and relieves users from spending a lot of time and effort to understand the GridSim toolkit code.

Key words grid simulator; simulation framework; workflow; XML

1 概述

网格模拟器为评估网格环境中各类任务调度算法的性能提供了一个可重复的、可控的实验环境。当前主流的网络模拟器有 Bricks, MicroGrid, SimGrid, GridSim, ChicSim, EDGSim, GridNet, OporSim 等。其中, GridSim^[1]为网格资源调度算法的研究提供了良好的基础, 因此, 本文选择 GridSim 工具包为研究对象。几大主流网络模拟器的比较见文献[1-2]。

研究人员需要花一定的时间熟悉复杂的 API 或源码, 然后通过编码完成模拟实验, 但这些对于模拟器的使用者都是重复性的过程。为了提高模拟器的易用性, 减少研究人员做模拟实验的编程量, 文献[3]用 Java 实现了一个基于 GridSim 模拟器的 GUI 网格环境建模工具 VM(Visual Modeler)用于生成部分实验代码。文献[4]使用 GridSim 工具包实现了一个基于 Web 的网格模拟调度平台 WGridSP。这些工具或平台虽然在一定程度上减少了编程的代码, 但其可扩展性差, 没有为用户自定义调度算法提供一个易扩展、易实现的公共接口。在网格环境中, 大型任务通常都是以网格 workflow 作业的形式来提交和执行的, 因此, 工作流调度成为网格调度中的一个研究重点。但是在当前的模拟器和模拟框架(平台)中, 仅 SimGrid^[5]支持网格工作流模拟。并且 SimGrid 是基于 DAG (Directed Acyclic Graph)的集中式调度的模拟。因此, 本文提出并实现了一个基于 GridSim 的网格模拟框架 GSF(GridSim Simulation Framework), 新增了网格工作流调度模拟。它为研究人员提供了一种易使用、可重用、易扩展的模拟环境。

2 GSF结构

GSF 以 GridSim 工具包为基础, 主要包含 2 个部分: 网

格模拟环境的构建和调度算法的模拟, 其框架结构如图 1 所示。其中, XMLParser 模块用于完成 XML 描述信息和 XML 配置信息的分析, 然后根据相应的信息建立相应的模拟实体。网格调度器部分用于模拟作业调度算法, 其中的 WorkFlowEngine 用于模拟工作流调度。

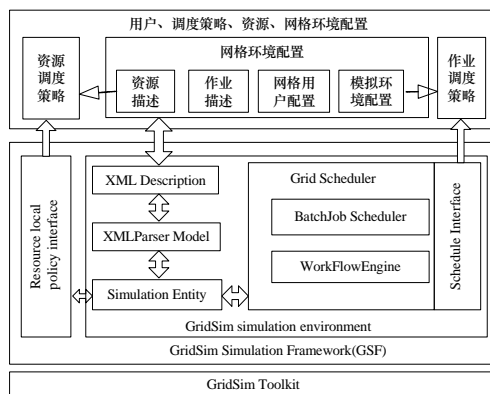


图 1 GSF 结构

3 GSF的设计与实现

GSF 用 XML 来描述资源、作业、用户 QoS, 并提供了

基金项目: 国家自然科学基金资助项目(60673165, 60433020)

作者简介: 胡志刚(1963-), 男, 教授、博士后、博士生导师, 研究方向: 并行计算, 网络计算; 李林, 硕士研究生

收稿日期: 2009-03-29 **E-mail:** leatinfy@gmail.com

带格式的: 缩进: 首行缩进: 18 磅, 右侧: 18 磅

一个网格调度器用于模拟调度算法。为了支持工作流的调度模拟,在调度器中实现了一个 WorkFlowEngine 来模拟工作流调度。

3.1 网格模拟环境的构建

在 GridSim 模拟网格环境中主要有 3 类实体: 资源, 作业和用户。本文用 XML 语言描述这 3 种实体, GSF 将读取 XML 描述信息并创建相应的实体。

3.1.1 资源

每个网格资源有资源名、处理单元(PE)数目、计算处理能力(MI/s: 每秒执行百万指令数)、通信带宽、价格等属性。为支持更多资源属性, GSF 对资源的属性进行了扩展。在 GridResource 类中添加了一个 HashTable 来保存扩展的属性, 使用方法 getExAttribute(String key)可获得相应的属性值。扩展资源属性如下, 在 XML 描述文件中, 节点<extensible-attributes>的子节点<attribute>表示扩展资源的属性:

```
<extensible-attributes>
  <attribute name=" trust">0.86</attribute>
  <attribute name="database">oracle</attribute>
</extensible-attributes>
```

3.1.2 作业

每一个作业实体代表一个独立作业, 包含了作业所需的处理器数目、作业长度(即作业所含的指令数量)、作业的输入和输出等属性。与资源的属性扩展类似, 在 extensible-attributes 节点中可以对作业属性进行扩展, 扩展的规则与资源属性扩展规则相同, 见 3.1.1 节。

3.1.3 工作流

为了描述工作流, 本文定义了一个工作流描述语言 GSWDL(GridSim Workflow Description Language), 其文档结构如下:

```
<workflow name="wf1">
  <Tasks>
    <Task id="1">
      <gridlet>
        <length>3000</length>
        <filesize>400</filesize>
        <outsize>500</outsize>
      </gridlet>
    </Task>
    <Task id="2">
      <gridlet>
        <length>3000</length>
        <filesize>400</filesize>
        <outsize>500</outsize>
      </gridlet>
      <Dependency operator="AND">
        <Parent state="FINISHED">1</Parent>
      </Dependency>
    </Task>
  </Tasks>
</workflow>
```

Workflow 节点包含一个 wgridlets, wgridlets 节点包含所有工作流子作业, 每一个工作流子作业用 Dependency 节点指定其依赖关系。定义了类 WorkFlow 表示一个工作流, 类 WGridlet 表示一个工作流中的子作业。类 WorkFlowCfg 读入一个工作流描述文档产生一个工作流实例。

当用户提交一个工作流作业给调度器时, 调度器用

WorkFlowEngine 执行工作流作业, 其结构如图 2 所示。WorkFlowEngine 在执行工作流调度时为工作流的每个子作业建立一个 WTaskManager。WorkFlowEngine 中有一个容器 WTMContainer 用来保存 WTM(WorkFlow Task Manager)的信息, 这样工作流调度器就可以与每一个 WTM 进行通信。

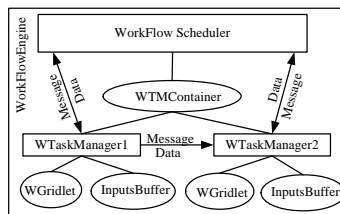


图 2 WorkFlowEngine 结构

从图 2 中可以看出 WTM 之间有单向通信, 并且 WTM 与工作流调度器之间也有通信。由于工作流子作业之间存在依赖关系, WTM 需要等待它的所有前驱作业执行完毕且传输数据到达时才能执行, 并且将作业的输出数据发送给它所有的后继作业。在工作流执行完成后, 工作流调度器将执行结果返回给用户。

3.1.4 网格用户

在 GSF 里设计了 GridEndUser 用于表示一个网格用户, 用于模拟一个网格用户的动作(提交网格作业)。用 XML 语言描述模拟网格用户的信息; jobs 节点中的 gridletfile 节点用于指定用户所拥有的作业 XML 文件。jobs 节点中有一个 type 属性指定作业类型, 其值可以是 BATCHJOBS 或 WORKFLOW。节点 QoS 用来指定用户的 QoS 要求。

3.2 调度算法模拟

调度算法模拟是 GridSim 的核心, 主要有 2 类: 资源本地调度策略和网格作业调度策略。

在 GridSim 中定义了 AllocPolicy 抽象类为资源本地调度策略的基类。在 GSF 中, 定义了 SchedulePolicy 抽象类作为作业调度算法的基类, 其类结构如图 3 所示。对于工作流作业, 分为静态调度和动态调度。进行静态调度时, 需要在 schedule 方法中为每个子作业绑定静态资源, 然后调用 workFlowSubmit 操作。对于动态调度, 则需要为每个工作流子作业绑定一个资源选择器 RS(ResourceSelector), 再用 workFlowSubmit 提交工作流。

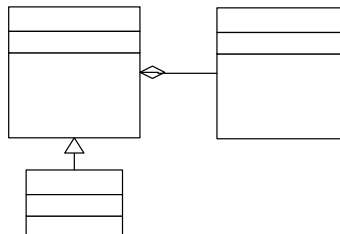


图 3 作业调度策略接口

在 GSF 中实现了一个网格作业调度器来模拟网格作业调度算法。作业调度器有 2 种: 批作业调度器和工作流作业调度器。当用户提交作业到网格中时, 由网格作业调度器接收用户的作业, 再根据作业类型执行相应的调度器调度到网格资源上执行, 最后调度器将作业执行结果发送给用户。在这个模拟过程中, 调度器调度网格作业到资源的过程就是执行

带格式的: 缩进: 首行缩进: 18 磅, 右侧: 18 磅

调度算法的过程。

3.3 类说明

本文设计了类 XmlParser 用于分析 XML 文件,读取 XML 文件的基本信息。所有的 Cfg 类都继承自同一个 XmlCfgReader 抽象类, XmlCfgReader 的所有子类通过实现 getCfgValue 方法来完成相应的 XML 信息读取。GridEvn 类调度相应的 Cfg 类来获取并初始化整个模拟环境。Scheduler 类和 Dispatcher 类用于处理网格作业。WorkFlowEngine 类完成 workflow 作业的调度。

4 应用实例

本节以一个使用 GSF 进行模拟的实例介绍如何用 GSF 快速地进行模拟实验,内容包括:如何使用 GSF 描述资源、网格用户和作业;如何实现网格调度策略和如何配置网格模拟环境。

(1)资源描述

实验中使用了表 1 所示的 5 个资源。

表 1 资源列表

名称	特征属性	MI/s	CPU	GS/S	时区
R0	Compaq, UNIX	515	4	64.37	+10
R1	Intel, P4 2 GHz, Linux	684	13	125.66	+10
R2	Origin 3200, IRIX	410	16	82.00	+2
R3	Origin 3200, IRIX	410	6	82.00	+2
R4	Sun, Ultra, Solaris	377	8	94.25	-5

资源 R0 的 XML 描述如下:

```
<resource>
  <name>R0</name>
  <arch> Compaq</arch>
  <os> UNIX </os>
  <time_zone>10.0</time_zone>
  <cost>64.37</cost>
  <baud_rate>100.0</baud_rate>
  <policy>TimeSharedWithReliability</ policy>
  <Machines>
    <Machine>
      <PES>
        <PE>
          <MIPSRating>515</MIPSRating>
        </PE>
      </PES>
    </Machine>
  </Machines>
  <extensible-attributes>
    <attribute name="reliability">0.96</attribute>
  </extensible-attributes>
</resource>
```

其中, TimeSharedWithAvailability 表示资源的本地调度策略。R0 的可用性属性通过扩展属性 availability 来指定,可用性值范围为[0, 1]。

(2)网格用户和作业描述

实验中模拟了 2 个用户 User0 和 User1,作业来源于 GWA (Grid Workloads Archive)。User0 拥有 10 个独立作业; User1 拥有一个 workflow 作业,用如下 XML 语言来描述这 2 个用户,其中, QoS 节点中指定了 User0 的截止时间要求:

```
<GridUsers>
  <User>
    <name>user0</name>
    <baud_rate>100.0</baud_rate>
    <Jobs type="BATCHJOBS">
```

```
<gridletsfile>./gridlets.xml</gridletsfile>
</Jobs>
<QoS>
  <Argument name="deadline">9396.0</Argument>
</QoS>
</User>
<User>
  <name>user1</name>
  <baud_rate>100.0</baud_rate>
  <Jobs type="WORKFLOW">
    <workflowfile>./workflow.xml</workflowfile>
  </Jobs>
</User>
</GridUsers>
```

(3)网格作业调度策略

实验中对批作业实现了用户代价最优的批调度策略类 TimeOptimisationScheduler; 对 workflow 作业实现了工作流子作业期望完成时间最短的动态调度策略类 ShortestExpTimeScheduler。

(4)网格模拟环境的配置

GSF 将所有的网格模拟信息存放在 config.xml 配置文件中。实验模拟中的网格资源、网格用户及网格调度器的相关信息都在配置文件中配置,实验中配置的网格信息如下:

```
<configuration>
  <grid-environment>
    <scheduler BaudRate="1000.0">
      <workflow_scheduler type="dynamic">
        ShortestExpTimeScheduler
      </workflow_scheduler>
      <batchjobs_scheduler>
        CostOptimisationScheduler
      </batchjobs_scheduler>
    </scheduler>
  </Resources>
  <Resource>./cfg/res1.xml</Resource>
  <Resource>./cfg/res2.xml</Resources>
</Resources>
<Users>
  <User>./cfg/guser.xml</User>
</Users>
</grid-environment>
</configuration>
```

(5)实验结果

模拟实验结果为: User0 workflow 作业的 makespan=339.644 ms, starttime=244.64 ms, endtime=584.284 ms。User1 workflow 作业的 makespan=99.14 ms, starttime=1 984.8 ms, endtime=2 083.94 ms。图 4 显示了 workflow 作业的执行时间。

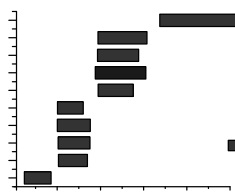


图 4 workflow 作业的执行时间

(下转第 40 页)

删除的内容:

(上接第 37 页)

从图 4 中可以看出,作业 A 的完成时间与作业 B、C、D、E 的开始执行时间有个时间间隔,这是因为 A 与作业 B、C、D、E 之间有数据传输,存在一个通信时间。

5 结束语

本文利用 GridSim 包实现了一个网格模拟框架 GSF,将传统的编写 Java 代码构建模拟环境的方式转变成完全基于 XML 描述和配置的方式,同时对 GridSim 进行了扩展,实现了一个 workflow 模拟器

WorkFlowEngine,并以应用实例的形式展示了如何使用 GSF 完成模拟实验。下一步工作是提供 GUI 工具以方便用户对 GridSim 进行了扩展,实现了一个 workflow 模拟器

参考文献

[1] Sulistio A, Cibej U, Venugopal S, et al. A Toolkit for Modelling and Simulating Data Grids: An Extension to GridSim[J]. Concurrency and Computation: Practice and Experience, 2008, 20(13): 1591- 1609.

[2] 卢鹏,金海,谢夏,等.关于网格模拟器的研究[EB/OL]. (2005-03-31). http://www.chinagrid.net/dvnews/upload/2005_03/05033100314923.doc.

[3] Sulistio A, Yeo C S, Buyya R. Visual Modeler for Grid Modeling and Simulation(GridSim) Toolkit[Z]. (2005-03-26). <http://www.>

带格式的: 居中

带格式的: 缩进: 首行缩进: 18 磅, 右侧: 18 磅

(上接第 37 页)

从图 4 中可以看出,作业 A 的完成时间与作业 B, C, D, E 的开始执行时间有个时间间隔,这是因为 A 与作业 B, C, D, E 之间有数据传输,存在一个通信时间。

5 结束语

本文利用 GridSim 包实现了一个网格模拟框架 GSF, 将传统的编写 Java 代码构建模拟环境的方式转变成完全基于 XML 描述和配置的方式,同时对 GridSim 进行了扩展,实现了一个 workflow 模拟器 WorkFlowEngine,并以应用实例的形式展示了如何使用 GSF 完成模拟实验。下一步工作是提供 GUI 工具以方便用户用 XML 对网格模拟环境进行配置和描述。

参考文献

- [1] Sulistio A, Cibej U, Venugopal S, et al. A Toolkit for Modelling and Simulating Data Grids: An Extension to GridSim[J]. Concurrency and Computation: Practice and Experience, 2008, 20(13): 1591-1609.
- [2] 卢 鹏, 金 海, 谢 夏, 等. 关于网格模拟器的研究[EB/OL]. (2005-03-31). http://www.chinagrid.net/dvnews/upload/2005_03/05033100314923.doc.
- [3] Sulistio A, Yeo C S, Buyya R. Visual Modeler for Grid Modeling and Simulation(GridSim) Toolkit[Z]. (2005-03-26). <http://www.chinagrid.net/grid/paperppt/gridsimu/g7.pdf>.
- [4] Kang Oh-han, Kang Sang-seong. A Web-based Toolkit for Scheduling Simulation Using GridSim[C]//Proceedings of the 5th International Conference on Grid and Cooperative Computing. Changsha, China: [s. n.], 2006.
- [5] Casanova H. SimGrid: A Toolkit for the Simulation of Application Scheduling[C]//Proceedings of the 1st IEEE/ACM International Symposium on Cluster Computing and the Grid. Brisbane, Australia: IEEE Computer Society Press, 2001.

编辑 张 帆

————分节符(下一页)————