异构分布式嵌入式系统自动化软硬件划分工具项目总结

目录

1 引言

1.1 项目简介

异构分布式嵌入式系统自动化软硬件划分软件由华东师范大学软件工程学院教育部软硬件协同设计技术与应用工程研究中心开发。

异构分布式嵌入式系统的出现大大提高了嵌入式系统的性能，现在被广泛应用于各种领域，如航空航天系统，自动驾驶系统等。在异构分布式嵌入式系统中，自动合理地将任务分配到分布式嵌入式系统的不同模块中，并对模块中的任务进行软硬件分配能够有效提高系统的性能。本软件根据异构分布式嵌入式系统优化设计方法，设计不同的算法对系统任务进行模块分配和模块内任务的软硬件划分。该软件会根据用户输入的系统通信关系矩阵、任务属性和系统约束，对该指定异构分布式系统按照希望的求解目标进行模块化和软硬件划分，最终给出最优的系统任务划分策略。该软件能充分使用系统软硬件的资源和能效，得到高性能、低成本的优化设计方案，极大地提高系统设计的合理性和效率。

1.2 项目背景

为应对日益增长的嵌入式系统性能需求，现代嵌入式系统向着异构化和网络化的方向发展：由多个异构平台联合，在硬件基础上辅以数个软件实现智能控制，并且软硬件紧密耦合，以此构造低功耗、高性能、低成本的专用系统。但传统的系统设计开发方法周期长，开发成本高，且难以针对软硬件的高耦合性做出合理优化。因此，如何快速有效地进行异构分布式嵌入式系统的软硬件自动化划分是如今嵌入式行业面临的巨大挑战之一。

软硬件划分方法是异构分布式嵌入式系统自动化设计和开发最基本的方法和技术之一。相较于传统设计方法，软硬件划分能够总和分析异构系统的功能、性能、功耗及成本需求，依据系统软硬件功能性能的要求以及现有资源（如 IP 核、软件构件等），使系统软件与硬件之间的并发性达到最优化，以此协调设计整个系统的体系架构。通过此种方法，可以避免独立设计软件和硬件带来的弊端，充分利用现有软硬件资源来提供系统性能、缩短系统开发周期，并降低系统开发成本。

本项目开发了一款针对异构分布式嵌入式系统的软硬件划分工具。在已提出的异构分布式嵌入式系统优化设计方法的基础上，对具体的算法进行实现，并开发出相应的工具供异构分布式系统设计人员使用。该工具会根据用户输入的系统任务关系、通信代价、任务属性和系统约束，对指定的异构分布式系统进行模块化和软硬件划分，最终给出最优的系统任务划分方案。该工具能充分运用系统软硬件的资源和能效，得到高性能、低成本的优化设计方案，极大地提高系统设计的合理性和效率。

1.3 项目的创新点

1) 优化异构分布式系统设计：该软件能充分运用异构分布式系统软硬件的资源和能效，得到高性能、低成本的优化设计方案，极大地提高系统设计的合理性和效率。

2) 约束和优化目标可灵活调整：我们提出的优化设计方法，在进行任务分配时，不仅考虑各处理器的性能约束，还会对处理器间通信资源和时间消耗进行优化。同时，该优化设

计方法的约束和优化目标可灵活调整。

3) 适用于多种场景：根据不同场景需求，该优化设计方法可针对系统的时间、能耗、成本等不同性能给出优化设计方案。

2 任务概述

2.1 目标

1. 通过文件给出的任务通信图和输入的模块数量，以通信代价最小化为目标，使用算法，将通信图划分为几个模块在分布式系统中运行

2. 对每个模块内的任务依据性能指标进行软硬件划分，使各模块在满足异构设备资源约束的前提下运行时间最短

2.2 运行环境

与软硬件划分软件相关的开发及运行环境如下表所示：

表 1 程序开发及运行环境

类 型	相关软硬件参数（推荐配置）
开发平台	Eclipse 2020-06 WindowBuilder 1.9.3 JDK 11.0.8
开发语言	Java
软件运行环境	Windows 10 JRE
运行硬件条件	CPU：Intel(R) Core(TM) i5-2400 或以上，主频在 2.10GHz 或以上； 内存：1GB 以上； 硬盘：50M 以上空余空间。

3 软件特点

1. 该工具实现了用户界面，界面简单明了，易于操作，可以读取用户指定的数据文件并输出最优模块化结果和软硬件划分结果
2. 在对异构分布式嵌入式系统模块化过程中，有五种算法可供选择，从中选出最优的算法
3. 软硬件划分工具纯自主，确保划分工具的独立性，本工具的开发完全掌握核心技术
4. 工具内功能模块间耦合程度低，容易加入新的功能，便于增量迭代开发。同时增强各模块之间的可协作性。便于后期对工具进行长期的维护、优化和迭代
5. 工具在进行任务划分时，不仅考虑各处理器的性能约束，还会对处理器间通信资源和时间消耗进行优化。同时，该优化设计方法的约束和优化目标可灵活调整，用户可以在三

个求解目标中选择一个：在系统开销和硬件面积约束下，系统运行时间最短；在系统运行时间和硬件面积约束下，系统开销最小；在系统运行时间约束下，硬件面积最小

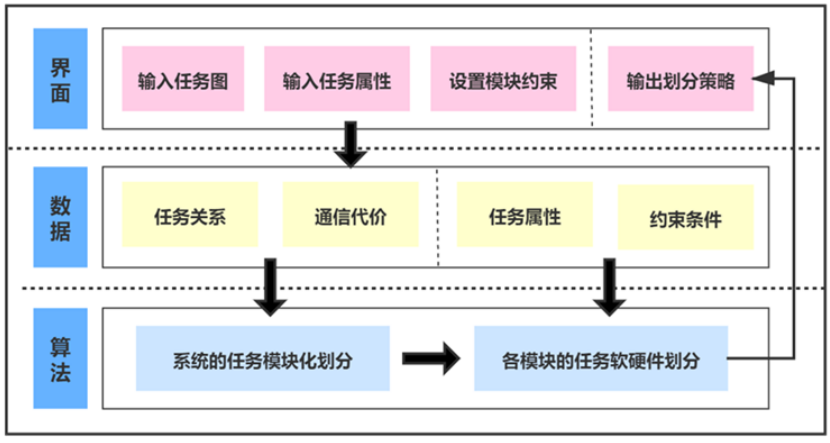
4 运行流程

1. 读取文件内的 3 个信息，任务数量 N，通信矩阵，任务属性
2. 模块划分有三个算法 MMM 算法（包含 3 种情况），KL 算法和 Modularize 算法，一共可以得到 5 种不同的模块化结果
3. 软硬件划分对 5 个模块化结果使用线性规划的方法，对模块内的任务进行软硬件划分，分 3 种求解目标
   在系统开销和硬件面积约束下，系统运行时间最短
   在系统运行时间和硬件面积约束下，系统开销最小
   在系统运行时间约束下，硬件面积最小
4. 比较结果，对 5 个模块化结果经过软硬件划分得到的求解目标绘制柱状图并进行比较，输出最优结果

5 总体设计

5.1 软件架构及流程

工具整体架构大致分为三层，从低至上分别为算法层、数据层和界面层。算法层主要实现对异构分布式嵌入式系统任务进行模块化划分和软硬件划分；数据层包含四种数据：任务关系、任务属性、通信代价和约束条件；界面曾主要负责实现与用户的交互，包括参数输入和结果输出。用户通过界面向算法层传递数据，算法运行之后将结果由界面输出给用户。



我们将整个软件系统分为 8 个子模块，分别是读取文件模块，显示文件信息模块，显示通信矩阵模块，读取模块数量模块，选择求解目标模块，读取约束信息模块，绘制不同算法结果柱状图模块和输出求解结果模块。整体的架构模块图如图 1 所示。整个软件运行的整体流程如下：

1) 读取文件信息：用户先将数据文件存放在软件同级目录，并输入文件名读取数据文件。

2) 显示文件信息：显示读取到的任务参数，并加上相关信息。
3) 显示通信矩阵：因为任务数量可能很大，对应的通信矩阵也会很大，放在界面中显示不合适，因此专门设置了一个按钮可以弹出通信矩阵窗口。
4) 读取模块数量：用户可以根据自己的需求输入划分的目标模块数量
5) 选择求解目标：用户可以根据自己的需求选择求解目标，一共 3 个求解目标可供选择
6) 读取约束信息：用户可以根据自己的需求按照选择的求解目标输入约束信息
7) 绘制不同算法结果比较：软件根据不同算法的求解结果输出柱状图，方便用户观察
8) 输出求解结果：软件通过比较不同算法的求解结果，输出最优算法的软硬件划分信息。

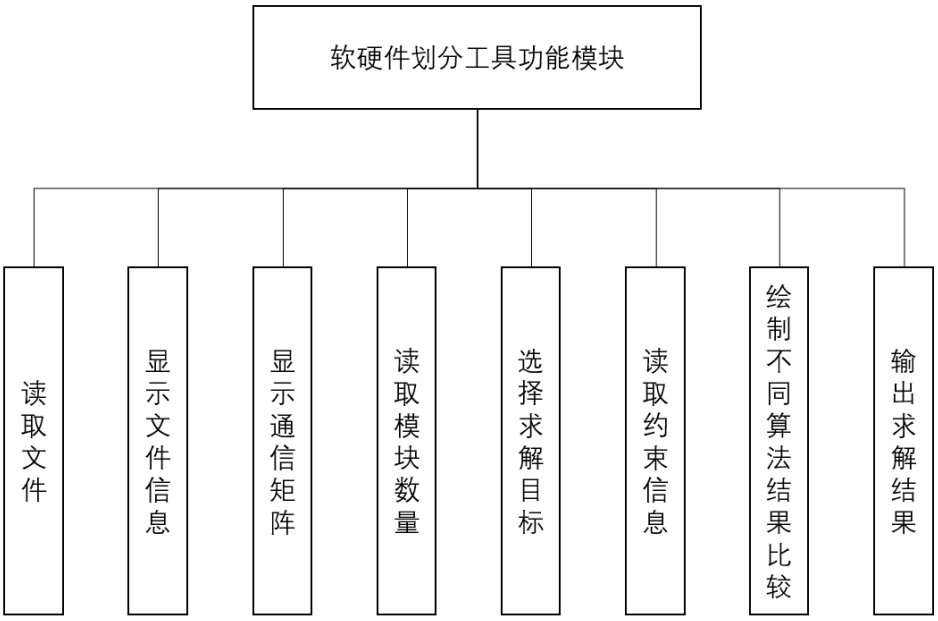


图 1 软件模块架构图

5.2 软件的系统建模

依据软件系统的模块结构设计以及功能流程图，我们对整个软件进行建模。模型能否足够精确描述我们的系统是实现软件设计的关键。统一建模语言（Unifed Modeling Language，UML）有其广泛的适用性，因此我们将 UML 作为模型的搭建软件。

- 用例图

用例图描述的是系统提供的功能模块，用例图在系统模块级的抽象层次上刻画了系统的功能需求之间的关系化及功能需要和用例之间的关系。

本系统软件的系统用例图如图 2 所示。用例是软件的使用者，使用者能够使用的相关的模块是输入文件模块，产看文件信息模块，查看通信矩阵模块，输入模块数量模块，选择求解目标模块，输入约束信息模块，获得结果比较柱状图模块和获得最终结果模块。

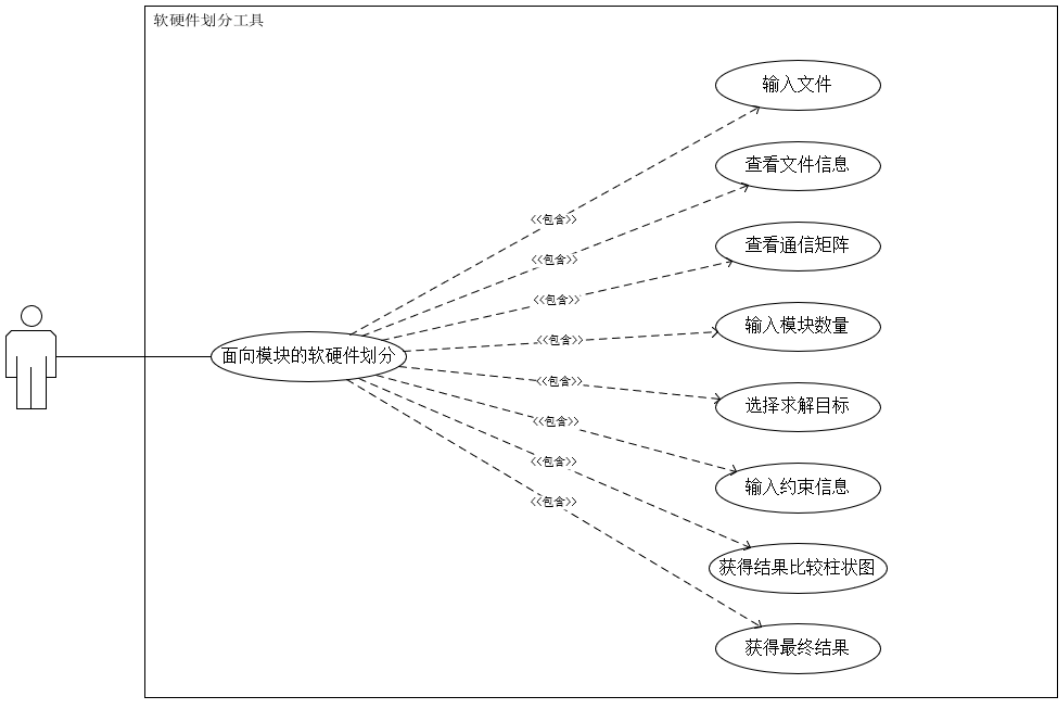


图 2 系统用例图

- 序列图

序列图，也称为时序图。主要用来更直观的表现各个对象交互的时间顺序，将体现的重点放在以时间为参照，各个对象发送、接收消息，处理消息，返回消息的时间流程顺序。本软件的时序图如图 3 所示。

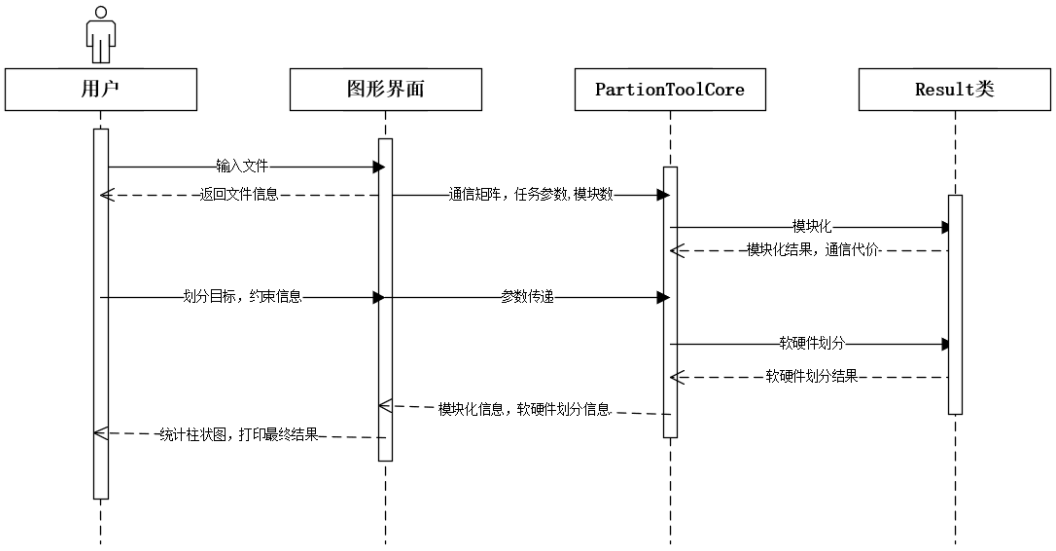
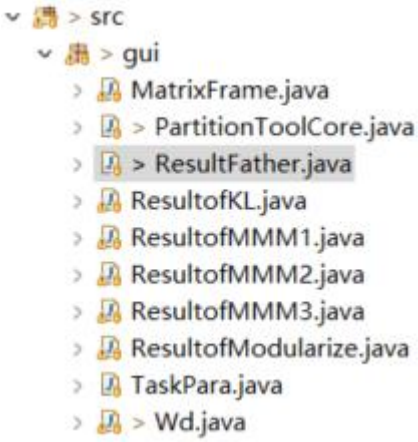


图 3 序列图

5.3 详细设计

工程目录结构：



软硬件划分软件主要包括三个部分：1. Wd.java 是 Swing 编写的 UI 界面；2. PartitionToolCore 是整个软件的核心，负责从界面里读取数据，调用处理数据的类和输出数据; 3. Result 类里面是模块化和软硬件划分两个方法。其中 ResultFather 是其它 5 个 Result 类的父类，具体模块化方法由 5 个子类根据 5 个不同的算法重写。本节将详细介绍每个部分的设计。

5.2.1 界面设计

本软件使用 Swing 来设计整个系统的 UI 界面，Swing 是 Java 的可视化 GUI 设计框架，可以帮助我们提高开发图形界面程序的速度。软件界面负责获取和显示用户的输入和输出，同时也控制了整个程序运行的逻辑流程。

软件的整个界面的布局设计如图 4 所示，主要分为左右两部分。左侧是用户与软件交互的地方，操作的时候从上往下，从左往右，可根据提示操作。右侧为结果显示部分，上半部分为输出柱状图的地方，下半部分为输出最终结果信息的地方。

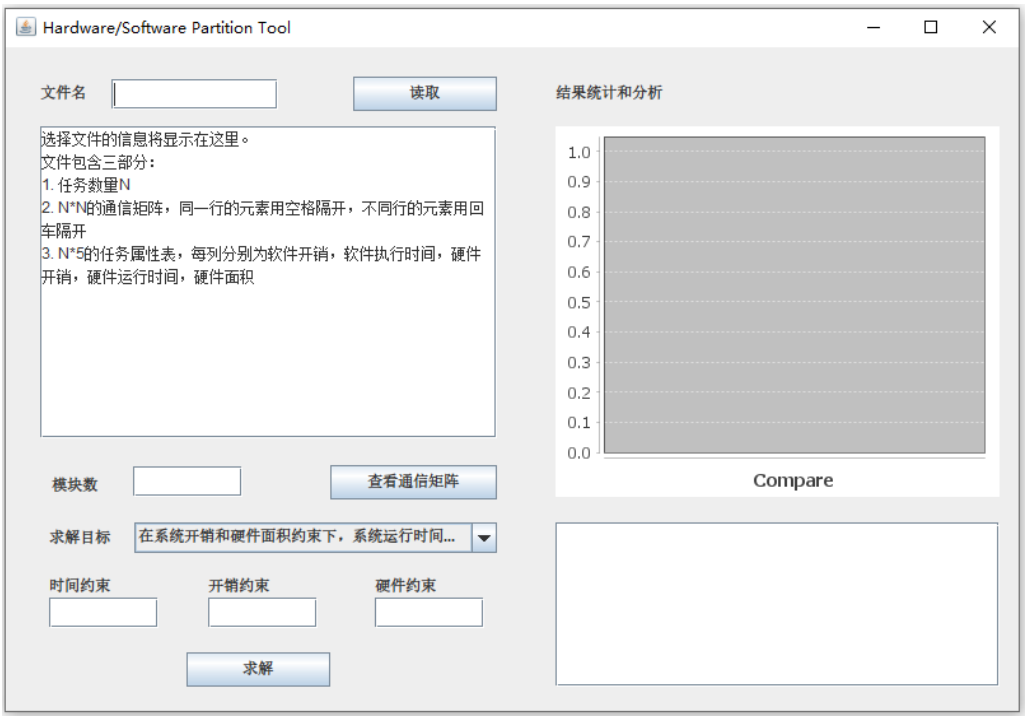
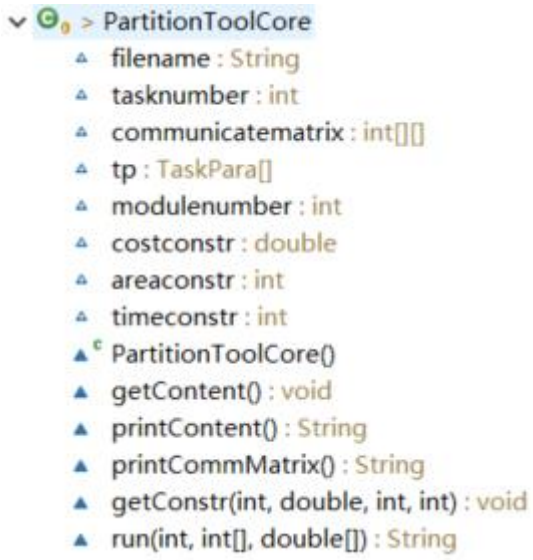


图 4 系统界面

5.3.2 PartionToolCore

PartionToolCore 的成员变量和成员方法如图所示



成员方法说明：

PartionToolCore(): PartionToolCore 的构造方法

getContent(): 被 UI 界面调用，根据输入的文件名 filename 将任务数量 tasknumber,通信矩阵存入成员变量 communicatematrix 中，任务参数存入成员变量 tp 中。

printContent(): 将读取到的文件数据中的任务属性输出在界面上

printCommMatrix(): 将读取到的文件数据中的通信矩阵输出在新的窗口中

getConstr(): 被 UI 界面调用，PartitionToolCore 获得用户选择的求解目标，并将约束信息传入模块数 modulenumber, 开销约束 costconstr, 面积约束 areaconstr, 时间约束 timeconstr。

run(): 在用户点击 UI 界面的求解之后，将调用 Result 类中的方法获得求解信息，并将信息输出在 UI 界面上。

5.3.3 Result 类算法设计

ResultFather 类的结构：

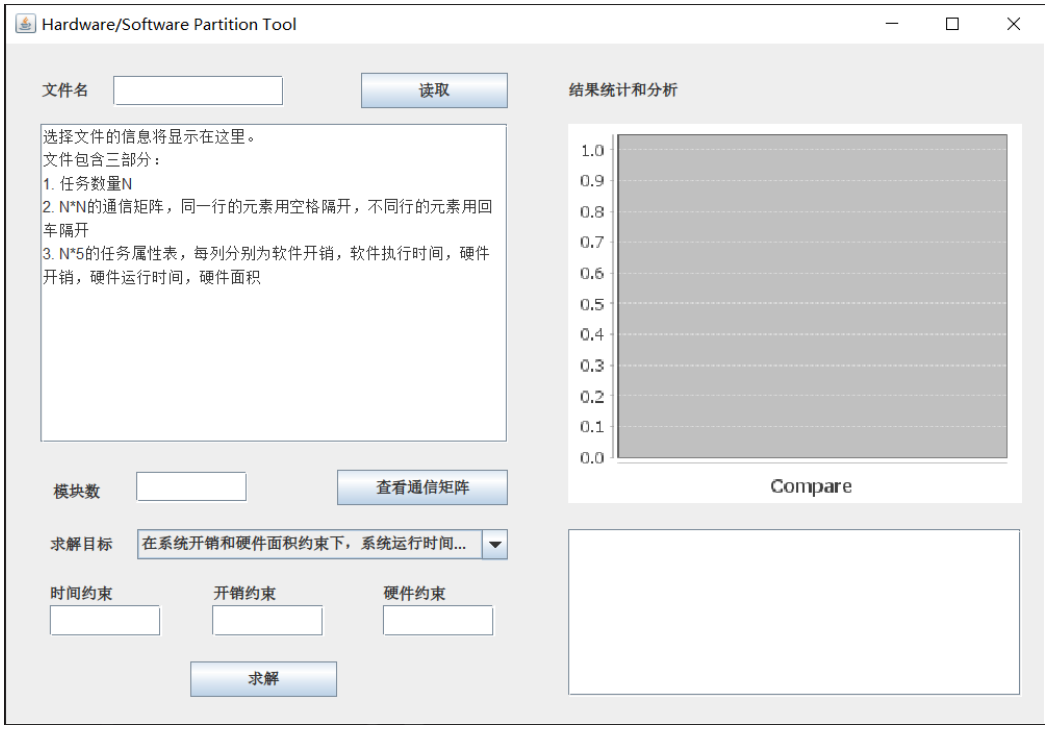
ResultFather(): 构造方法

hwswPartitionX(): 针对不同求解目标使用线性规划算法的软硬件划分方法，内部调用了mintimeX()

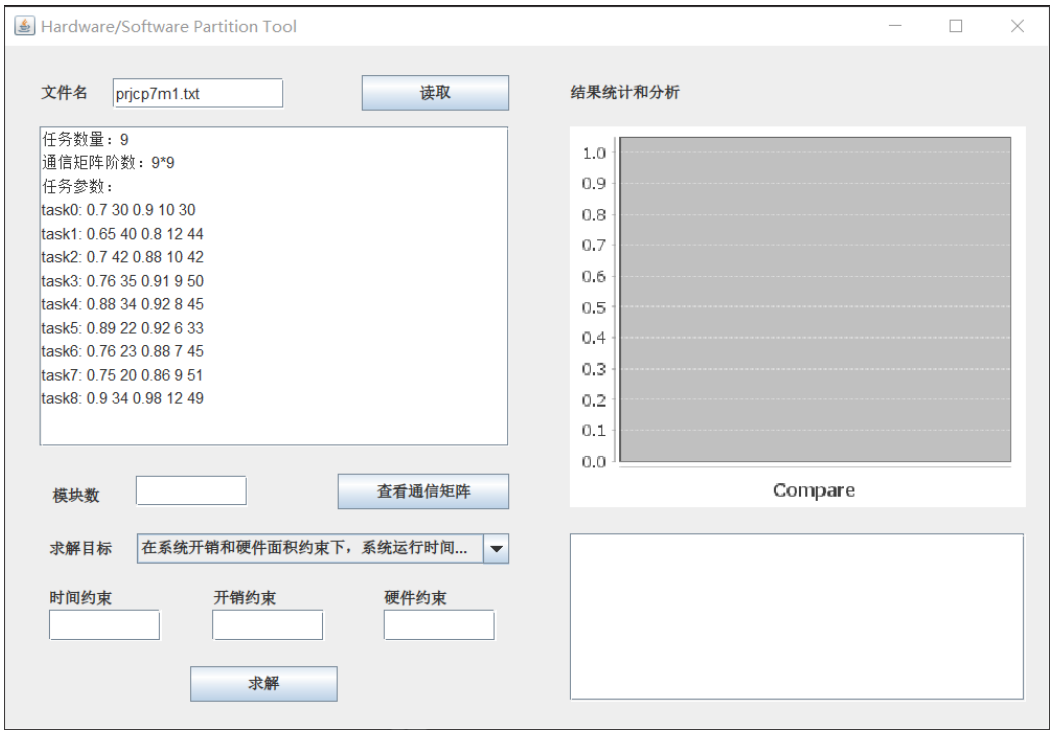
modular(): 模块化方法，5 个子类对该方法进行重写。ResultofMMM1, ResultofMMM2, ResultofMMM3 这 3 个子类使用的是基于聚类的模块化算法，层次聚类算法在初始时，将每个任务都视为一个单独的簇，然后不断将相似度大的簇合并到一起，直到簇数满足划分模块数的要求，每个簇对应一个模块。ResultofKL 使用的是基于 KL 算法的模块化算法，KL 算法是 1970 年由 Kernighan 提出的划分算法，用于将一张边上有权值的图中的节点划分为若干个子集，并使子集间总距离最短。在这个子类中 KL 算法用于将任务间存在通信代价的系统划分为若干个模块，并使模块间通信代价最小。ResultofModularize 使用的是基于贪心的模块化算法，该算法尽可能将通信代价大的任务分配到同一模块中。

6 使用说明

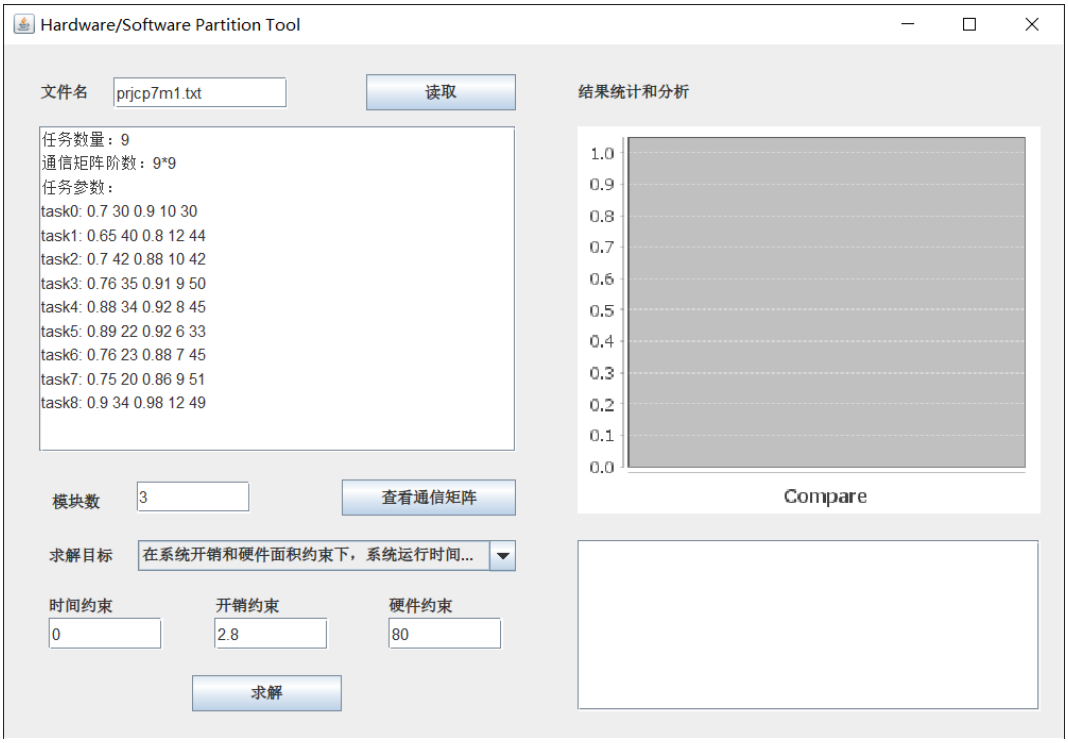
1. 电脑需要安装 JRE，双击 PartitionTool.jar，打开图形界面



2. 将数据文件放在软件同级目录下，输入文件名，点击读取，显示文件信息



3. 求解目标一共有 3 个可以选择，这里为第一个为例即在系统开销和硬件面积约束下，系统运行时间最短。输入模块数 3，时间约束在该求解目标下用不到填 0，开销约束填 2.8，硬件约束填 80



4. 点击求解，得到不同算法的统计结果和最优结果