

综合模块化航空电子多约束分区调度方法^{*}

杨骏峰 李 峭

(北京航空航天大学电子信息工程学院 北京 100191)

摘要: 在综合模块化航空电子系统(integrated modular avionics, IMA)中,采用严格的空间和时间分区管理(partitioning)保证同一个模块上运行的不同应用可以共享处理资源。根据系统硬件资源和应用的要求,定义分区的多类型约束条件,使得分区到在各个模块上的分配和时分访问调度成为复杂的组合优化问题。通过将分区分配的预处理与满足性模理论(satisfiability modulo theories, SMT)求解调度表的方法相结合,可以减少断言式和分区调度时刻变量的数量,提高求解效率;其中,预处理过程采用最大独立团算法,随后将剩余的分区约束条件转换成 SMT 工具可识别的逻辑表达式,形式化求解得到各个分区的调度时刻。通过规模不同的算例,验证了该方法可行性,并说明预处理过程对于快速判断满足性要求和缩短求解时间的好处。

关键词: 调度; 航空电子; IMA; 分区约束条件; 图论; SMT 工具

中图分类号: TN915 **文献标识码:** A **国家标准学科分类代码:** 510.5030

Multi-constrained partitioned scheduling method of integrated modular avionics

Yang Junfeng Li Qiao

(Beihang University, Beijing 100191, China)

Abstract: In the integrated modular avionics (IMA), strict space and time partitioning management ensures that different applications running on the same module can share processing resources. According to the requirements of system hardware resources and applications, the multi-type constraints of partition are defined, which makes the partitioning and time-division access scheduling become complex combinatorial optimization problems. By combining the pre-processing of partition allocation with satisfiability modulo theories (SMT) to solve the scheduling table, it is possible to reduce the number of variables at the time of assertion and partition scheduling, and improve the efficiency of the solution. The preprocessing process uses the maximum clique problems algorithm, and then transfer the rest of the district constraints into the logical expression that SMT can identify, using the formal solution to get the time of each partition scheduling. The feasibility of this method is verified by different numerical examples, and the advantage of the preprocessing process for quickly judging the satisfaction requirement and shortening the solution time is illustrated.

Keywords: scheduling; avionics; IMA; constraints of partition; graph theory; SMT tools

1 引言

随着微电子和计算机科学的发展,在航空电子领域采用共用处理资源的综合化形式,综合模块化航空电子系统(integrated modular avionics, IMA)的概念应运而生^[1]。在 IMA 系统中采取 ARINC 653 规范定义的核心操作系统、分区操作系统和统一的应用层接口中,形成空间分区、时间分区的调度方式^[2],用以尽力避免共享处理资源之后,

航空电子系统的不同功能构件之间,或者不同关键级别的应用软件之间在空间、时间上相互影响。

IMA 系统中,每一个处理单元叫做模块,通过硬分离约束提供航电功能的共享和可依赖的通信。分区是对应用程序进行的一个抽象,是一种具有特定功能的应用的组合。分配模块中的每个分区分配一定的执行时间,严格的按照各个分区的周期循环执行。一个强壮的分区系统要满足不同分区在同一个模块中执行不会在空间和时间上相互影

收稿日期:2016-12

^{*} 基金项目:中央高校基本科研业务费专项资金(YWF-15GJSYS-055)项目资助

响。因此在分区的时间调度上,就提出了需要满足的多种约束条件。之后将分区调度抽象成为复杂度很高的组合优化问题进行求解。

SMT 算法已经被用于解决 TTE 网络层和任务层的静态调度问题,由于 TTE 实时网络作为 IMA 系统的通信传输网络。因此可以将二者调度方法进行统一,提供了一种求解 IMA 系统分区调度的问题的方法。并可结合图论^[4]理论进行预处理^[5],优化基于 SMT 算法的分区调度。传统解决分区调度这类 NP 完全(NP-complete)问题,采用启发式算法,能够减少求解的时间,但是,这种整合为线性方程组(MILP)方法,忽略了调度的严格周期性^[6],并且需要调用 IBM ILOG CPLEX 求解器。本文的贡献在于,引进 SMT 求解算法,可以有针对性地将 IMA 分区调度约束条件的断言式(assertion)转换为满足性问题的逻辑运算,使得在保持 SMT 求解^[7]的严谨性的条件下利用先验的系统配置信息降低求解规模。

2 IMA 系统分区调度模型

2.1 IMA 系统结构^[3]

ARINC653 规范中给出了一种 IMA 系统结构模型,主要由 ARINC653 应用程序接口层、操作系统层、硬件接口层、硬件层、模块层及 AFDX 网络组成,如图 1 所示。

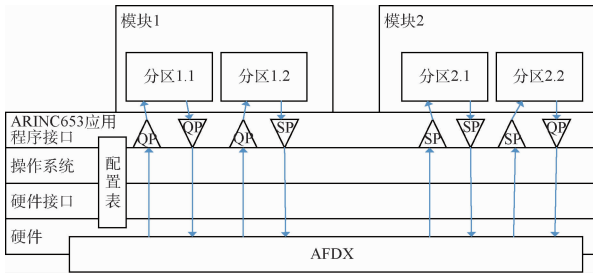


图 1 IMA 系统结构

1) 模块层(应用软件层):由各个分区组成,每个模块可以宿主一个或者多个分区,执行应用程序,满足飞机航电功能。

2) ARINC653 应用程序接口层:主要为航空电子模块提供服务,可以实现应用之间的通信。通过定义的两端端口:列队端口(QP)和采样端口(SP),分别传递不可分割消息和可分割/重组消息。保证满足不同要求分区间的单向通信。

3) 操作系统层:管理各个分区和单个分区内部应用进程之间的通信。操作分区间通信系统与硬件接口层相互作用,可以实现通信调度、内存管理和健康监控。

4) 硬件接口层:包括借口驱动器,并可提供对内存但愿和时钟的访问。

5) 硬件层:提供各种外围设备接口:如 AFDX 总线等^[8],用来保证模块间分区调度的实时性通信。

2.2 分区调度模型定义

分区空间 $\Pi = \{\pi_1, \dots, \pi_n\}$ 和模块空间 $P = \{p_1, \dots, p_m\}$ 。

模块模型:对于每一个模块 $p_j \in P$,其属性包括可利用的记忆空间容量 M_j 和可宿主最大分区数量 K_j 。

分区模型:对于每一个 $\pi_i \in \Pi$,其属性包括分区周期 T_i ,分区时间预算 b_i ,即分区的执行持续时间,分区记忆预算 m_i ,即分区所占用的记忆空间。

2.3 分区调度约束

对于分区的约束条件,主要包括模块硬件属性上与通信软件上两大类,具体如下:

C1:每个分区只能存在于一个模块中

定义分区模块向量组 $(a_{i,j})$,其中 i 代表分区编号, j 代表模块编号,则:

$$a_{i,j} = \begin{cases} 1, & \text{如果分区 } \pi_i \text{ 被分配到模块 } p_j \\ 0 & \end{cases}$$

也就是说:

$$\sum_{p_j \in P} a_{i,j} = 1, \forall \pi_i \in \Pi \quad (1)$$

C2:每个分区的执行时刻必须满足严格周期性

$$t'_i = t_i + (k-1)T_i \quad (2)$$

C3:每个分区必须在所在模块的主时间帧(major frame, MAF)内执行。主时间帧的长度为该模块中所有分区周期的最小公倍数。

$$0 \leq t'_i \leq \text{MAF} - b_i \quad (3)$$

C4:每一个模块内,不同分区的每一个执行时间不能重叠分区 i 的第 m 个分区的执行时间为:

$$I_m(t_i) = [t_i + (m-1)T_i, t_i + (m-1)T_i + b_i], \text{ 则:}$$

$$\forall (\pi_i, \pi_k) \in \Pi^2, \forall m, n \in N^+, \forall p_j \in P$$

$$a_{ij} = a_{kj} = 1 \Rightarrow I_m(t_i) \cap I_n(t_k) = \emptyset \quad (4)$$

C5:每一个模块内,所有分区的记忆空间之和不能超过模块的记忆空间容量

$$\sum_{\pi_i \in \Pi} a_{i,j} m_i \leq M_j, \forall p_j \in P \quad (5)$$

C6:每一个模块内,分区的各数之和不能超过模块可宿主的最大分区数量

$$\sum_{\pi_i \in \Pi} a_{i,j} \leq K_j, \forall p_j \in P \quad (6)$$

2.4 利用图论算法进行预处理

分区调度问题的复杂度随着完全断言式和分区调度时刻变量数量的增加成指数型增长,分区调度前减少断言式和变量数量,就可以缩减整个调度问题求解的运行时间。由于分区在模块分配过程中,存在兼容性约束 C4。因此可构建分区无向图,抽象成图论中的最大团(maximum clique problem)问题^[9],利用回溯求解算法进行预处理。

首先,构建分区无向图 $G(V, E)$,以分区对应图的节点集合 V ;图中的边集合 E 的确定规则如下:根据分区约束条件 C4,将可以共存于同一模块的两个分区相连,这样图中

之间没有连线的两个节点,代表这两个分布一定不能共存于同一个模块。

随后,定义图 G 中的最大的无关集,即:这个无关集的每一个元素则必不能共存于同一个模块。利用无向图矩阵 G ,调用回溯求解算法,求得最大独立团。

最后,将无关集元素在计算分区调度前,分配于不通模块,大大减少了分区调度时刻的变量数量,加快了计算效率。

例如图 2 所示,是一个 6 个分区元素组成的无向图,相连分区可以并存于模块,因此可以构造无向图矩阵 G :

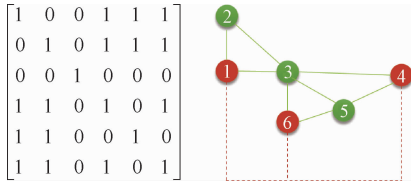


图 2 无向图及其矩阵

利用图论回溯求解求得最大无关集为[1,4,6]。因此,可以事先分配分区 1,4,6 到不同的模块之中,这样我们可以把问题分配变量从 6 个缩减到 3 个,对于 NP 完全问题,大大提高了运行效率。

对于求解满足条件的解模型。传统方法是将约束条件整合为最大整数线性方程组(MILP)进行求解。本文基于可满足性模理论(satisfiability module theories, SMT)来解

决分区调度问题。

3 利用 SMT 对分区调度进行求解

3.1 SMT 求解逻辑公式

SMT^[10]求解工具实际上是完备的,以回溯为基础的算法,用于解决合取范式逻辑的可行解空间,是布尔可满足问题 SAT(boolean satisfiability, SAT)的一种扩展。SAT 工具只能解决包含布尔变量的逻辑命题。如 $\neg a \wedge (a \vee b)$, 其中 a, b 均为逻辑变量。而 SMT 典型的理论包括固定规模的实属类型变量、数组、线性运算等。例如 $(a - b < 5) \wedge a > 7$, 其中 a, b 是实属变量。利用 SMT 这一优点,可以对分区调度解模型的计算提供方便。无需将分区的约束条件整数线性化求解,只需要将约束条件进行转换为 SMT 可读的形式化表达,再交给相应的求解工具即可。在这里选取 Yices 作为逻辑公式的求解工具。

3.2 SMT 求解分区调度问题

在应用中使用的 SRI 开发的 SMT 求解器 Yices 工具,该求解器对逻辑公式的输入有其自身规定的形式化语言规范。例如示例 $(a - b < 5) \wedge a > 7$, 需要转化为 $(and(<(-ab)5)(>a7))$ 的形式。所以利用 SMT 的求解过程如图 3 所示。

利用 SMT 求解分区调度时刻表的主要流程包括:设置分区调度时刻变量,将每个分区调度时刻变量带入到所有约束当中,生成 1 组完备的形式化断言集,对其进行格式

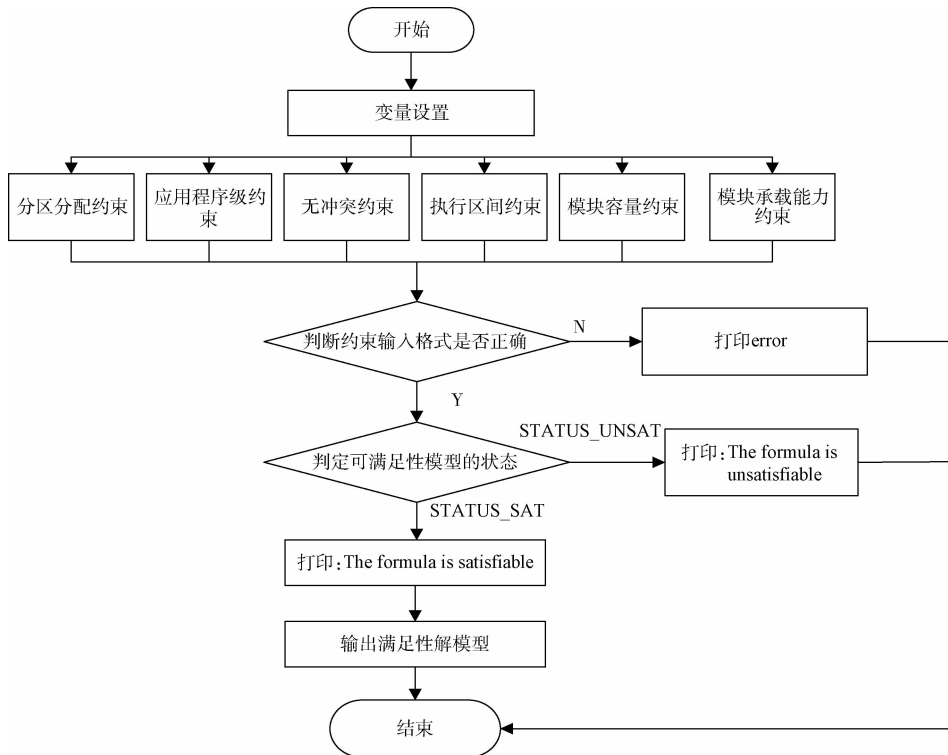


图 3 SMT 求解流程

后解析,再判断是否有满足性解模型,得到相应的满足约束条件的各个分区的模块分配和调度时刻。

4 案例研究与对比分析

4.1 SMT 求解可行性验证

可行性验证主要包括预处理与非预处理两种方法,流程如图4所示,给出的IMA分区调度案例为6分区、三模块。分区周期[50,200],分区执行时间[10,60],模块容量为3。

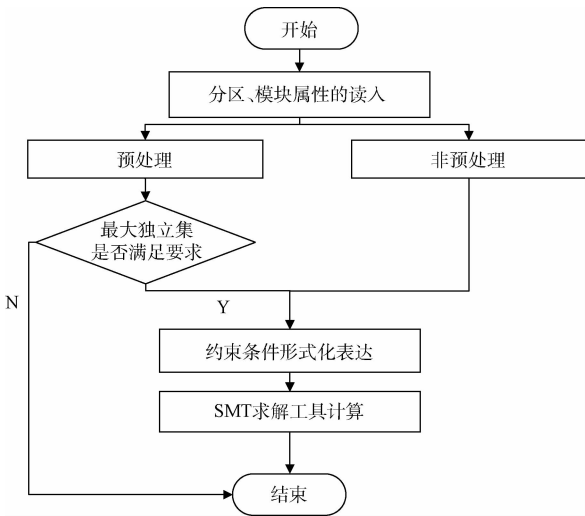


图4 分区调度流程

4.2 预处理与非预处理速率对比

增加分区数量与模块数量,对两种方法SMT求解方法进行运行速率对比数据如图,实验的硬件环境为Intel i5 3.20 GHz,内存为8.00 GB,运行结果如表1所示

表1 运行结果

实验组编号	有无满足性解模型	模块数量	分区数量	执行时间(非预处理)/ms	执行时间(预处理)/ms	运行速度比率
A	有	3	6	172	117	1.47
B	有	4	8	580	316	1.83
C	无	3	8	336	6	—
E	有	6	12	11 245	2 595	4.33
F	有	10	20	583 859	116 531	5.01

从ABEF对比试验数据中,不难看出,在有满足性解模型时,进行预处理后的计算速度要快于非预处理方法,且在随着航电规模的扩大,运行速率提高越明显。从BC组对比实验数据中看出,在无满足性解模型时,进行预处理可以在调用SMT进行求解前就终止计算,大大的提高了效率,避免了浪费不必要的时间。

5 结论

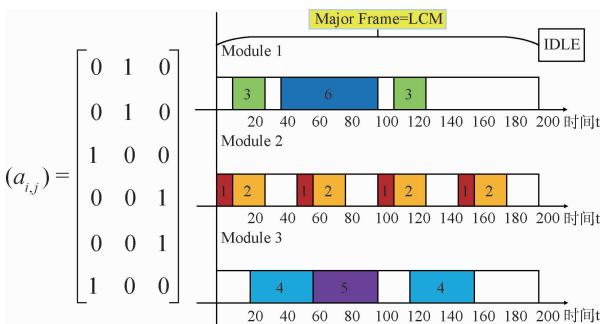
本文针对综合模块化航电系统的结构特性和ARINC653分区调度机制,根据其硬件结构和应用软件的特性提出的分区调度约束条件。采用不同于传统的MILP求解方法,验证了SMT工具进行分区调度的可行性,以及根据分区无向图,求解最大无关集的预处理的优化方法,大大的提高了SMT进行求解的效率。为大规模的IMA系统分区调度提供可能性。

对于分区间通信的AFDX网络调度,仍需进一步进行验证研究。对于不同SMT求解软件的运行效率也需要进一步的对比。

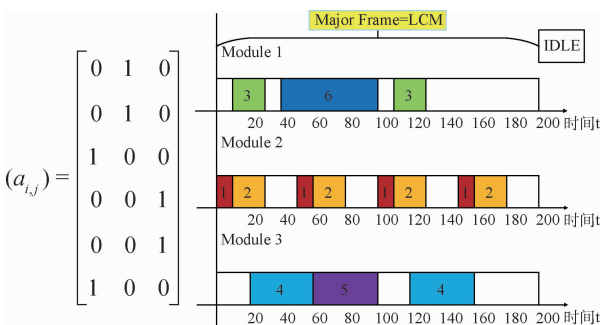
参考文献

- [1] 熊华钢,周贵荣,李峭.机载总线网络及其发展[J].航空学报,2006,27(6):1135-1144.
- [2] 乔乃强,徐涛,谷青范. ARINC65分区调度算法的研究与改进[J]. 计算机工程,2011,37(20):249-251.
- [3] 徐文. 综合模块化航电系统分区鲁棒性测试[C]. 2014航空试验测试技术学术交流会议论文集,2014.
- [4] SHARIEH A, AL RAWAGEPFEH W, MAHAFAZH M, et al. An algorithm for finding maximum independent set in a graph[J]. European Journal of Scientific Research, 2008, 23(4): 586-596.

非预处理得到的分区模块矩阵 $(a_{i,j})$ 及调度表



预处理得到的分区模块矩阵 $(a_{i,j})$ 及调度表



通过得到的实验结果,可以看出利用SMT方法求解IMA系统多约束分区调度是可行的。

(下转第160页)