

概率模型检测在动态能耗管理中的应用

杜伊 何洋 洪玫

(四川大学计算机学院 成都 610065)

摘要 如何平衡嵌入式设备的能耗和性能表现,成为了一个热门话题。动态能耗管理是一种在保证系统性能的基础上降低其能耗的有效方法,其关键点是如何生成有效的动态能耗管理策略。在概率模型检测技术的基础上,提出了一种生成和验证动态能耗管理策略的方法。首先对目标系统和能耗管理目标建模,然后利用 PRISM-games 工具进行动态能耗管理策略的合成,同时利用模型检测工具 PRISM 对合成的动态能耗管理策略进行验证。实验表明,该方法具备可行性和有效性。

关键词 动态能耗管理,概率模型检测,策略合成,策略验证

中图分类号 TP301 文献标识码 A DOI 10.11896/j.issn.1002-137X.2018.01.046

Application of Probabilistic Model Checking in Dynamic Power Management

DU Yi HE Yang HONG Mei

(College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract It has become a hot topic to make a trade-off between energy consumption and performance of embedded devices. Dynamic power management (DPM) is an efficient way to reduce the devices' energy consumption while guaranteeing its' performance, and the key point of DPM is the DPM strategies. Based on probabilistic model checking, a method to generate and verify DPM strategies was proposed. The target system is modeled as SMGs models, goals are set as rPATL properties, and then the probabilistic model checking tool (PRISM-games) is used for strategies synthesis aiming at danamic energy management. Furthermore, PRISM is used for verifying the synthesized strategies. The experimental results show that the method is feasible and efficient.

Keywords Dynamic power management, Probabilistic model checking, Strategies synthesis, Strategies verification

1 引言

嵌入式系统以及便携式设备等近年来不断涌现,但其有限的能源供给或过度的能源消耗,使得节能成为一个炙手可热的话题。研究者们提出了很多有关节能的方法,动态能耗管理(Dynamic Power Management, DPM)是其中一种有效的方法^[1]。DPM 被广泛用于低能耗系统的设计中,可以使工作组件在开和关的状态之间进行转换,或在几种不同能耗开销下的状态之间转换,同时关注系统性能和能耗之间的平衡。

早期的 DPM 策略主要是超时策略,该策略简单易实现,但是在实际系统中效果并不十分理想。经过发展,后来出现了诸如随机策略、预测策略、机器学习等策略,这些策略各有利弊。

本文基于概率模型检测(Probabilistic Model Checking)^[2]技术,提出了一个合成(Synthesis)^[3]和验证 DPM 策略的方案。在方案中,首先对目标系统和能耗管理目标建模,

然后利用 PRISM-games^[4]工具进行动态能耗管理策略的合成,同时利用模型检测工具 PRISM^[5]对合成的策略进行验证。实验表明,该方法具备可行性,能够针对不同的能耗管理目标合成相应的能耗管理策略;同时,合成的策略满足约束条件,具备有效性;最后与传统的 MAPLE 求解策略进行对比,结果表明本文方案生成的策略更接近策略目标。

2 基于概率模型检测的动态能耗管理策略合成

模型检测^[6]是一种重要的形式化验证方法,通过穷举搜索待验证系统模型的状态空间,实现对系统的验证。概率模型检测在模型检测的基础上,通过建立对状态空间转化的概率,可以针对带有随机行为或不确定性行为的系统进行定量分析。合成^[3]是模型检测工具提供的一种新功能,它针对非确定性系统,给定一个特性性质,可以提供一种策略(或控制器)来解决系统中的非确定性问题,从而满足给定的性质。PRISM-games^[4]是由 Birmingham 大学和 Oxford 大学联合开

发的概率模型检测合成工具,它支持随机多竞争者博弈(SMGs)模型^[4]、带开销的概率交替时间时序逻辑^[4](rPATL)、模型检测和合成算法。本节介绍了抽象的DPM模型及策略合成的整体方案,然后对合成方案的各步骤进行了详细的说明。

2.1 动态能耗管理模型和策略合成方案

本节首先介绍抽象的动态能耗管理模型,将DPM的马尔科夫^[6]模型抽象为如下几个部分。

(1)服务请求者(Service Requester, SR):一个带有状态集 R 的马尔科夫链,描述了系统服务请求的到达。

(2)服务提供者(Service Provider, SP):一个带有状态 S 的可控马尔科夫链。它的状态代表了系统操作的模式(例如能耗状态)。它的转换是带有概率的,且该概率由能耗管理者控制。

(3)能耗管理者(Power Manager, PM):实现了公式 $f: S \times R \rightarrow A$,从服务请求者和服务提供者的状态集到可能的命令集 A 。这种公式是决策过程的抽象表示,能耗管理者观测系统状态和工作量,然后做出决定,并发出命令来控制系统的未来状态。

(4)开销指标(Cost Metrics):结合了 $S \times R \times A$ 下的能耗与性能指标。

策略合成方案首先将DPM模型表示为模型检测合成工具PRISM-games支持的SMGs模型,然后将需要达到的策略目标描述为PRISM-games支持的rPATL性质。将SMGs模型和rPATL性质输入到PRISM-games工具中,如果合成性质能被满足,该工具会搜索模型的状态空间并自动生成一个能满足该性质的路径,这条路径是工具求得的最优解,也就是它认为的满足目标的最好策略。如果合成性质不能被满足,工具返回不存在策略。该方法的主要流程如图1所示。

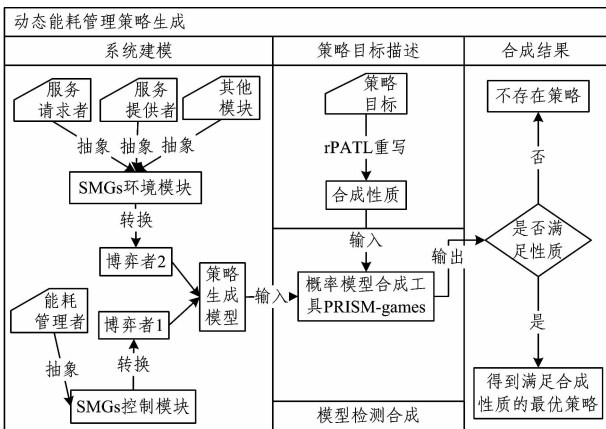


图1 基于概率模型检测的DPM策略生成方法的主要流程
Fig. 1 Main flow of DPM strategies synthesis method based on probabilistic model checking

SMGs本身是一个状态变换系统,对于每一个状态而言,它的变换是由博弈(game)中的一个博弈者(player)或概率决定的,在每个被控制的状态下,非确定性的选择由博弈者来决定。因此,本文将系统中的各模块抽象为不可控模块和可控

模块的双博弈者竞争模型(two-player games)。不可控模块,即本文所说的环境(environment)模块,对于一个DPM系统,服务何时到达、是否有存储空间储存待处理的服务、处理器处理服务所需的功率等都是不可控的因素;可控模块是指可以控制和改变的模块,即本文说的控制器(controller),例如如何控制系统在不同能耗状态下的转换,这一模块根据环境的变化动态地控制系统的运行。

其次,用时序逻辑性质rPATL表示要合成的策略目标,这类性质需要同时考虑能耗和性能两方面的因素。注意,rPATL允许博弈者集合有一个策略要么能确保一个有概率性的事件发生,要么能确保预期的rewards能满足某些阈值。

最终可以获得零个或多个策略,通过在策略下的一系列模拟,记录系统的策略指导,由此归纳出在不同环境状态下系统能耗的分配和变化。

2.2 动态能耗管理系统建模——SMGs模型

2.2.1 系统环境建模

由于SMGs是离散的模型,若要模拟时间这种连续型的变量,则需要将其离散化。本文建立了时钟模块和电池模块,模型如图2所示,当且仅当电池有电量时才能运行,否则所有模块均进入停止状态。因此,系统环境建模包括5方面的内容:时钟模块、电池模块、服务生成模块、服务请求队列模块、服务处理模块。

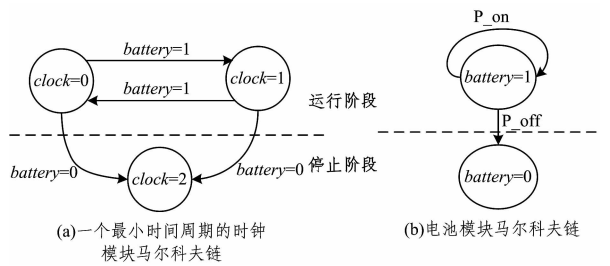


图2 时钟和电池模块的抽象模型
Fig. 2 Abstract model of clock and battery module

服务生成模块被用来提供服务,包括两个状态:空闲状态(idle)和请求状态(request)。空闲状态时等待服务的到达,在请求状态时得到服务并将服务传递给服务请求队列模块。将其抽象成一个三状态的马尔科夫链,该模块有状态集 $SR = \{sr, i = 0, 1, 2\}$, $P_request$ 表示有服务请求到达的概率, P_send 表示将服务送往服务请求队列的概率。模型如图3所示。

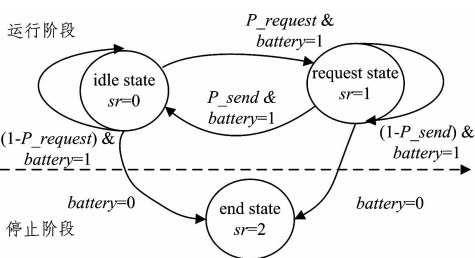


图3 服务生成模块的马尔科夫模型
Fig. 3 Markov model of service generation module

服务请求队列模块提供缓冲区来暂存未处理的服务,该模块有状态集 $Q = \{q, i=0, 1, 2, \dots, Q_{MAX}, Q_{MAX}+1\}$, Q_{MAX} 是服务请求队列的最大长度, sp 是服务处理模块中的状态变量,图 3 中,当 $sp=0$ 时才能处理服务,且服务处理完成的概率为 P_{serve} 。服务请求队列模块的状态变化主要受服务生成模块和服务处理模块的影响,接受来自服务生成模块的服务,并根据能耗管理模块的指示,将服务传递给服务处理模块,若服务请求队列模块的服务队列已满,继续向该队列传递的任务将会丢失,模型如图 4 所示。

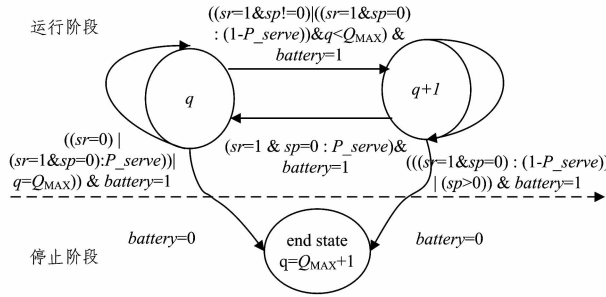


图 4 服务请求队列模块的马尔科夫模型

Fig. 4 Markov model of service request queue module

服务处理模块的状态集 $SP = \{sp, i=0, 1, \dots\}$, 该模块中可能包含两个或多个状态。可以是两个状态,通常是开(on)状态和关(off)状态;也可以是多个状态,如一种比较典型的四状态模型包括了忙碌状态(busy)、空闲状态(idle)、待机状态(standby)以及睡眠状态(sleep)。一般情况下,该四状态模型只有在忙碌状态下才能处理服务,因此忙碌状态下的能耗也是最高的,空闲、待机、睡眠状态都无法处理服务,并且能耗逐渐降低,但它们从该状态转换为忙碌状态所需的时间一般又是逐级增加的。该模块的抽象模型如图 5 所示。

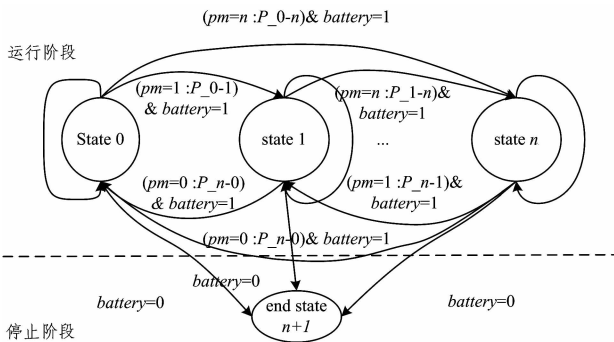


图 5 服务处理模块的马尔科夫模型

Fig. 5 Markov model of service processing module

2.2.2 策略生成模块建模

策略生成模块,即本文所说的控制器模块,是一个不确定的模块,其状态变化与服务处理模块息息相关,它指导服务处理模块状态的变化,同时它本身的状态也可以表示为服务处理模块的变化。状态集 $PM = \{pm, i=0, 1, \dots\}$, 其取值取决于服务处理模块。例如, $pm=0$ (pm 表示策略生成模块中能耗管理状态变量)表示命令服务处理模块进入活跃状态, $pm=1$ 表示命令服务处理模块进入空闲状态等,模型如图 6 所示。

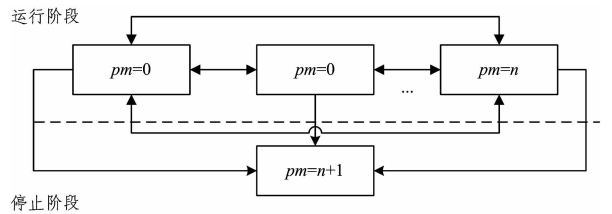


图 6 策略生成模块的抽象模型

Fig. 6 Abstract model of strategies synthesis module

2.2.3 Rewards 和其他模块建模

Rewards 是 PRISM 和 PRISM-games 提供的一个很重要的功能,它使得本文能够在模型运行时获取一些累积的或即时的指标,能在一定程度上评判策略的好坏。在模型中,最常用的 3 个考查指标是平均能耗、平均队列长度和服务丢失率,因为 rewards 变量只能记录累积的变量,所以需要平均值,还需记录总的时间。因此,对于以上 3 个考查指标,需要 4 个 rewards 变量,即总能耗(power)、总队列长度(queue)、总的服务丢失数(lost)、总时间(time)。

2.3 策略目标建模

在 DPM 系统的策略目标描述中,不仅需要普通的 rPATL,还需要多目标的 rPATL (Multi-Objective rPATL),因为策略目标至少需要考虑能耗和性能两方面因素,而性能又需要用多个指标来表示,所以本文所描述的多目标 rPATL 的基本公式如下:

$$\langle\langle controller \rangle\rangle (R_{\sim r_1} [F target1] \& R_{\sim r_2} [F target2])$$

$\langle\langle controller \rangle\rangle$ 表示对 controller 这个博弈者进行模型检测合成, R 是求解 rewards 属性的标识,“&”用来连接两个单一的性质。例如 $\langle\langle controller \rangle\rangle R \{“power”\} < 1000 [F bat = 0] \& R \{“lost”\} < 5 [F bat = 0]$ 表示需要同时满足当电池电量耗尽时能耗小于 1000,且当电池电量耗尽时服务丢失数少于 5 个。

除 R 操作符表示的目标之外,本文也可以用 P 操作符表示目标。 P 操作符可以求解性质的概率,例如 $\langle\langle controller \rangle\rangle P < 0.01 [G “down state”] \& \langle\langle controller \rangle\rangle R \{“power”\} < 1000 [F bat = 0]$, 该性质的前半部分表示进入“down state”这一状态的概率需要小于 0.01, 整句性质表示合成一个既满足进入“down state”状态的概率小于 0.01 又满足当电池电量耗尽时能耗小于 1000 的策略。

2.4 动态能耗管理策略的合成

对于传统生成策略方法,需要建立系统的转移矩阵,从而建立策略的线性最优问题,然后通过 MAPLE 符号求解器得到相应的策略。

不同于上述传统方法,本方案在策略生成模块中,通过模型检测工具的合成功能自动得到满足性质的最优策略,该过程是一个自动化的过程。只需将上述每个模块建立成正确的 SMGs 模型,并用 2.3 节中的方法描述想要满足的目标性质,将上述模型和目标性质输入到概率模型检测工具 PRISM-

games 中,如果存在满足该目标性质的最优解(策略),即可自动合成得到;若不存在,工具会返回 false。

3 基于概率模型检测的动态能耗管理策略的验证

本节将上一节生成的策略模型转换为 PRISM 支持的 DTMCs 模型,将需要验证的性质转换为概率计算树逻辑(Probabilistic Computation Temporal Logic, PCTL),再利用 PRISM 进行模型检测,从而得到结论。该方法的主要流程如图 7 所示。

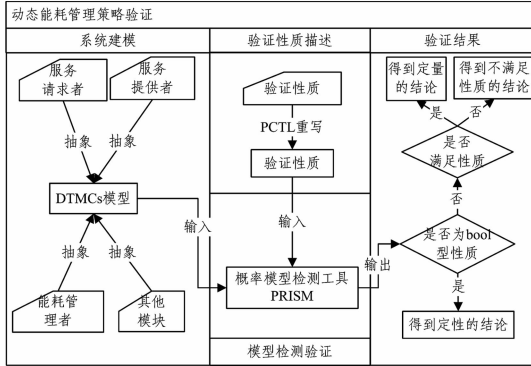


图 7 基于概率模型检测的 DPM 策略验证方法的主要流程

Fig. 7 Main flow of DPM strategies verification method based on probabilistic model checking

3.1 建立动态能耗管理系统的 DTMCs 模型

该模型不同于第 2 节的 SMGs 模型,不需要将它们划分为不同的博弈者,因为它们已经是完整的系统,只需要用 DTMCs 的语法完整且准确地表达出这几个模块即可。对于服务生成模块、服务请求队列模块和服务处理模块,应与 SMGs 模型一致,以保证两个模型的无误性。而能耗管理模块则大不相同,在策略生成模型中,用不确定的方式表达模块的行为,而在本节中则用确定的方式将它们表达出来。图 8 给出了某个策略下 pm 模块的部分代码示例。图 9 给出了本文构建的 DPM 验证模型的 DTMCs 的结构图。

<pre> module PM pm : [0..4]; //0 表示命令 SP 进入 active 状态 //1 表示命令 SP 进入 idle 状态 //2 表示命令 SP 进入 idlelp 状态 //3 表示命令 SP 进入 standby 状态 //4 表示命令 SP 进入 sleep 状态 //现有策略 </pre>	<pre> ... //由 PRISM-games 合成的策略 一约束 0.05 [tick] sp=9 -> (pm'=0); [tick] sr=0 & sp=0 -> 0.66183933; (pm'=0) +0.33816067; (pm'= 1); [tick] sp=1 sp=3 sp=6 -> (pm'=0); [tick] !(sp=9 (sr=0 & sp= 0) (sp=1 sp=3 sp= 6)) -> (pm'=pm); ... //一直处于 sleep 状态 [tick] pm=4 -> true; /一直处于 active 状态 [tick] pm=0 -> true; ... </pre>
---	--

图 8 一个 DTMCs 模型中 PM 模块的部分代码示例

Fig. 8 A piece of sample code of PM module in DTMCs model

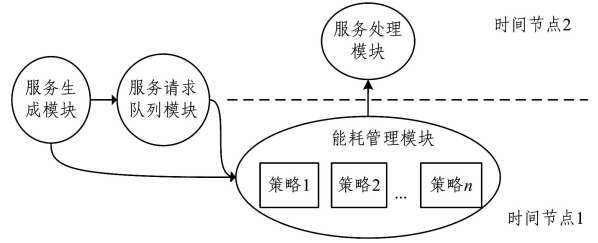


图 9 DPM 策略验证模型的 DTMCs 结构图

Fig. 9 DTMCs structure chart of DPM strategies verification

Rewards 模块不仅是策略生成模型的重要指标,也是策略验证模型的重要指标。与策略生成模型类似,在策略验证模块中本文也有 3 个基本的指标需要表述:能耗、服务队列长度和服务数量丢失数。这 3 个指标是本文参考和衡量策略好坏的重点。

需要重点指出的是,不止存在不同状态下处理器的能耗,还存在另一种不同状态之间的转换消耗——时间的消耗。比如,处理器有 4 个状态:活跃状态(active)、空闲状态(idle)、待命状态(standby)以及睡眠状态(sleep),处理器的能耗在这 4 个状态下各不相同,从一个状态转换到另一个状态还需要消耗时间,且时间长短也不相同,而且可以预见的是 active 和 sleep 之间的状态转换应该是最耗时的。因此,在建立能耗这个 Rewards 指标时,应该把这两种类型的消耗都构建出来。

3.2 策略验证性质描述

DTMCs 可验证的性质分为 3 种类型:基于概率的性质、基于稳定状态的性质和基于 Rewards 的性质。在对 DPM 策略进行验证的过程中,本文主要考虑能耗和性能指标,即基于 Reward 的性质。Rewards 性质主要分为 3 类:即时的 Rewards(Instantaneous Rewards)、累积的 Rewards(Cumulative Rewards)和可达的 Rewards(Reachability Rewards)。

即时的 Rewards 是指在某一时间节点 k 下的 Rewards 预期值,其基本公式如下:

$$R_{-r}[I=k]$$

累积的 Rewards 是指到达时间节点 k 后的 Rewards 总共累计的期望值,其基本公式如下:

$$R_{-r}[C \leq k]$$

可达的 Rewards 是指当到达某一状态时累积的 Rewards 期望值,其基本公式如下:

$$R_{-r}[F \varphi]$$

即时的 Rewards 可以表述诸如服务队列长度这类需要考虑即时数值的指标,累积的 Rewards 可以用于表述诸如能耗这类需要考虑累积开销的指标。

本文可以考虑 3 方面的性能能耗表现:1)随着时间的推移,能耗、服务丢失数和服务队列长度等需要对比的指标的变化情况;2)在某一指标满足条件后,能耗和服务丢失数等其他需参考指标的最后数值;3)某一个或几个指标满足既定条件时概率的大小。因为针对不同的 DPM 系统,对能耗和性能的要求是不同的,有的系统需要服务全部得以处理,而有的服务可以允许服务数量有个位数的丢失,而更看重能耗指标。

3.3 策略验证

将模型和性质输入到 PRISM 工具中即可进行模型检测并获得检测结果。PRISM 提供了两类检测(验证和实验),在性质中不存在未定义的变量时,使用验证获取检测结果;在性质中存在未定义的变量时,使用实验获得检测结果。图 10 和图 11 分别是 PRISM 的验证和实验结果的示例。

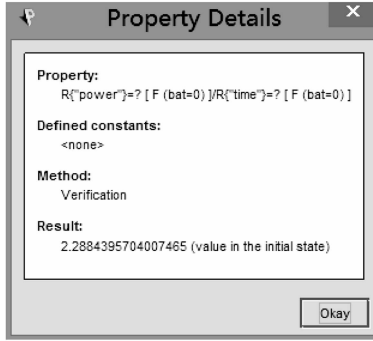


图 10 PRISM 验证结果示例

Fig. 10 A sample of PRISM verification result

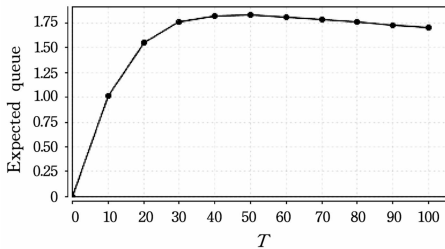


图 11 PRISM 实验结果示例

Fig. 11 A sample of PRISM experimental result

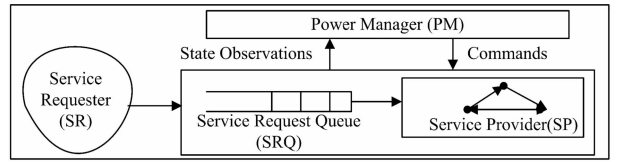


图 12 系统模型

Fig. 12 System model

该模型的 SP 模块共有 5 个状态:睡眠状态(sleep)、待命状态(standby)、空闲状态(idle)、低能耗空闲状态(idle low power, idlelp)和活跃状态(active),只有在 active 状态下设备才能进行数据的读写操作。SP 模块状态间的转换如图 13 所示。

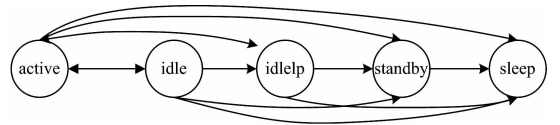


图 13 SP 模块能耗状态转换图

Fig. 13 Energy consumption state transition diagram

表 1 列出了在每个状态下系统的平均能耗和服务时间,表 2 列出了在每对状态之间进行转换所需的平均时间,数据来源于文献[1]。

表 1 每一状态的平均能耗和服务时间

Table 1 Average energy consumption and service time of different states

	sleep	standby	idlelp	idle	active
能耗/W	0.1	0.3	0.8	1.5	2.5
服务时间/ms	0	0	0	0	1

表 2 状态间的平均转换时间/ms

Table 2 Average conversion time between states/ms

	sleep	standby	idlelp	idle	active
active	—	1	5	220	600
idle	1	—	5	220	600
idlelp	5	—	—	220	600
standby	220	—	—	—	600
sleep	600	—	—	—	—

4.2 动态能耗管理策略生成方案的可行性验证

使用第 2 节中描述的方法,首先抽象系统模型,对每一个模块建立相应的马尔科夫模型,然后将其组合为 SMGs 模型,再描述策略目标建立性质,最后利用工具 PRISM-games 自动获得相对应的策略。

首先,对 PM, SP, CLOCK, SR, SRQ, BATTERY 模块构建 SMGs 模型,并定义两个博弈者(player1 和 player2), player1 表示控制模块,player2 表示环境模块。将 PM 模块归于 player1 中,将 CLOCK 模块、BATTERY 模块、SR 模块、SRQ 模块和 SP 模块归于 player2。

然后,建立 rPATL 性质。本实验选择的策略目标为:在给定了不同的平均服务队列长度下提供策略使能耗最小。该 rPATL 性质的基本格式如下:

$$\langle\langle\alpha\rangle\rangle(R\{power\})_{\min}[F\ target] \cap (R\{queue\}_{\text{constraint}}[F\ target]/R\{time\}[F\ target])$$

4 实验验证

本节设计了 3 个实验对方案进行验证。实验 1:验证方案是否能生成 DPM 策略,说明方案的可行性;实验 2:对实验 1 中生成的策略进行验证,说明方案的有效性;实验 3:与传统 MAPLE 求解生成的策略进行对比,说明本方案的优势。

4.1 实验环境和实验对象

实验硬件环境:CPU: Intel(R) Core(TM) i5-4200H CPU 2.80GHz; RAM: 4.00GB。实验的软件环境:操作系统: Microsoft Windows 8; 概率模型检测合成工具: PRISM-games 2.0; 概率模型检测验证工具: PRISM 4.3。

本文选取 IBM 的 Travelstar VP 硬盘^[1-7]作为实验的模型,如图 12 所示,该模型由 4 部分组成:服务请求模块(SR)、服务请求队列模块(SRQ)、服务处理模块(SP)和能耗管理模块(PM)。SR 模块为整个系统提供服务,从 SR 输送的服务直接到达 SRQ 模块,等待 SP 模块的执行,当 SRQ 模块的队列已满时,新到达的服务将会被丢弃,SP 模块负责处理服务,但它是在 PM 模块的控制之下进行的,PM 模块会根据情况的不同控制 SP 模块。

其中, α 表示 PM 模块所对应的博弈者, $R\{power\}$, $R\{queue\}$ 和 $R\{time\}$ 分别表示能耗、队列长度和时间的 Rewards 值。该性质的目标是使能耗最小, 且具有约束条件, 即平均服务队列长度小于或等于 $constraint$ 值。根据该性质, 利用 PRISM-games 的合成技术就可以合成满足上述两个目标的策略。最后通过改变不同的 $constraint$ 值自动获得不同约束条件下的策略。

将上述 SMGs 模型和 rPATL 性质输入到模型检测工具 PRISM-games 中, 就可以自动生成策略。

实验表明, 利用本文提出的方法, 策略被成功合成。图 14 给出了在某约束条件下生成的策略选择, 此图表示当 $sr=1$ 且 $q=1$ 时, 有 0.155432 的概率使 $pm=1$, 有 0.844568 的概率使 $pm=0$ 。表 3 总结了几种不同约束条件下 PRISM-games 合成的最优策略的完整策略选择。

Strategy choice	Module(action) (player)	Probability	Update
$0 \rightarrow 1, 0$	{SRQ} (1, SR)	0.155432	$c \leftarrow 1, pm \leftarrow 1$
$0 \rightarrow 1, 0$	{SRQ} (1, SR)	0.844568	$c \leftarrow 1, pm \leftarrow 0$
\times	{SRQ} (1, SR)	0.0	$c \leftarrow 1, pm \leftarrow 0$
\times	{SRQ} (1, SR)	0.0	$c \leftarrow 2, sp \leftarrow 1, sr \leftarrow 2, sr \leftarrow 0$

图 14 PRISM-games 生成策略的策略选择示例

Fig. 14 The synthesized strategies choice by PRISM-games

表 3 PRISM-games 合成的不同约束条件下的最优策略

Table 3 The optimal strategies synthesized by PRISM-games under different constraints

策略编号	约束条件	策略描述
1	平均服务队列长度小于或等于 1.5	模型开始运行时, 即使 SP 进入 active 状态; 当 SP 处于 active 状态, SRQ 队列长度不满且 SR 处于 idle 状态时, SP 以 0.256733 的概率进入 idle 状态; 当 SP 处于 active 状态, SRQ 队列长度不满且 SR 处于 request 状态时, SP 以 0.155432 的概率进入 idle 状态; 当 SP 不处于 active 状态且队列长度满时, SP 以 0.99905 的概率进入 active 状态。
2	平均服务队列长度小于或等于 1.0	模型开始运行时, 即使 SP 进入 active 状态; 当 SP 处于 active 状态且 SRQ 队列长度不满时, SP 以 0.586237 的概率进入 idle 状态或以 0.091143 的概率进入 idlelp 状态; 当 SP 不处于 active 状态且队列长度满时, SP 以 0.999985 的概率进入 active 状态。
3	平均服务队列长度小于或等于 0.5	如果 SP 模块处于 active 状态且 SRQ 模块的队模型开始运行时, 即使 SP 进入 active 状态; 当 SP 处于 active 状态且 SRQ 队列长度不满时, SP 以 0.436628 的概率进入 idle 状态或以 0.066931 的概率进入 idlelp 状态; 当 SP 不处于 active 状态且队列长度满时, SP 以 0.999985 的概率进入 active 状态。
4	平均服务队列长度小于或等于 0.1	模型开始运行时, 即使 SP 进入 active 状态; 当 SP 处于 active 状态, SRQ 队列长度为空且 SR 处于 idle 状态时, SP 以 0.63816 的概率进入 idle 状态; 当 SP 处于 idle 或 sleep 状态时, 使 SP 进入 active 状态。
5	平均服务队列长度小于或等于 0.05	模型开始运行时, 即使 SP 进入 active 状态; 当 SRQ 队列长度为空且 SR 处于 idle 状态时, SP 以 0.33816 的概率进入 idle 状态; 当 SP 不处于 active 状态时, 使 SP 进入 active 状态。

4.3 动态能耗管理策略的生成与验证方案的有效性验证

使用第 3 节中描述的方法, 首先抽象给定的模型, 对每一个模块建立相应的 DTMCs 模型, 然后抽象验证目标建立性质, 最后利用工具 PRISM 自动地获得相对应的验证结果。

首先, 使用 DTMCs 语义分别构造 CLOCK, BATTERY, SR, SRQ 和 SP 模块, 再根据上述实验生成的策略构建 PM 模块的 DTMCs 形式。

然后, 将验证目标建立为 PCTL 性质。本实验选取了 3 个评估指标, 即平均能耗、平均服务丢失数和平均服务队列长度。这 3 个验证性质的通用格式如下:

$$R\{reward\}=?[F battery=0]/R\{time\}=?[F battery=0]$$

根据上述性质即可求解得到不同 rewards(能耗、服务丢失数和服务队列长度)在电池电量耗尽后的平均值。

将上述 DTMCs 模型和 PCTL 性质输入到模型检测工具 PRISM 中, 即可根据性质利用概率模型检测得到验证结果。

表 4 列出了本文策略生成方案中获得的策略能耗与性能表现, 可以看出生成策略的实际平均队列长度都是满足约束条件的, 因此本文的策略生成方案能够获得满足策略目标的结果, 具备有效性。

表 4 本方案生成的各条策略的能耗与性能表现

Table 4 Different energy consumption and performance of different strategies generated by our scheme

策略编号	策略约束	平均能耗	平均服务丢失数	平均队列长度
1	≤ 1.5	0.8833	0.1139	1.4951
2	≤ 1.0	1.3056	0.0939	0.9933
3	≤ 0.5	1.6664	0.0031	0.4988
4	≤ 0.1	2.1822	0.0004	0.0997
5	≤ 0.05	2.2884	0.0002	0.0498

4.4 动态能耗管理策略生成方案的对比实验

通过系统的转移矩阵, 可以建立策略的线性最优问题, 然后通过 MAPLE 符号求解器得到相应的策略, 这是传统的生成策略的方法。本文利用 MAPLE 求解得到不同约束条件下的策略, 利用策略验证方案验证策略的性能与能耗, 并与表 4 中的数据进行比较, 结果如表 5 所列。

表 5 本方案与传统 MAPLE 求解方案的能耗与性能表现的对比

Table 5 Comparison of energy consumption and performance between our method and MAPLE

策略编号	策略约束	平均能耗(本方案/MAPLE 求解方案)	平均服务丢失数(本方案/MAPLE 求解方案)	平均队列长度(本方案/MAPLE 求解方案)
1	≤ 1.5	0.8833/0.9620	0.1139/0.1139	1.4951/1.4932
2	≤ 1.0	1.3056/1.3186	0.0939/0.0932	0.9933/0.9919
3	≤ 0.5	1.6664/1.6717	0.0031/0.0030	0.4988/0.4983
4	≤ 0.1	2.1822/2.1964	0.0004/0.0004	0.0997/0.0989
5	≤ 0.05	2.2884/2.2893	0.0002/0.0002	0.0498/0.0496

由表 5 可以看出, 通过两种方法得出的 3 个指标都比较接近。但由本方案得到的策略更接近它的约束条件, 虽然由本方案生成策略的平均服务丢失数略多于已有的策略, 但平均能耗也更小。由此对比实验可以得出本策略生成方案的两个优势: 1) 相较于传统方案建立最优问题并求解策略的方法, 本方案能根据模型和性质自动生成策略, 更简易; 2) 本方案生成的策略更接近于策略目标。

5 相关工作

目前, 国内外已经有很多针对 DPM 的研究, 如 Umair 等人^[8]提出了 DPM 系统策略生成的几种不同方法, 为概率模型检测分析 DPM 策略提供了基础; Luca 等人^[9]综述了 DPM 及其策略生成技术; Dhiman 等人^[10]利用机器学习算法在已

Computer Science, 2013, 40(7): 273-276. (in Chinese)

杜晓刚, 党建武, 王阳萍, 等. 基于萤火虫算法的互信息医学图像配准[J]. 计算机科学, 2013, 40(7): 273-276.

- [20] DENNIS E L, THOMPSON P M. Functional Brain Connectivity Using fMRI in Aging and Alzheimer's Disease[J]. NIH Public Access Author Manuscript, 2014, 24(1): 49-62.
- [21] CHEN W, YU Y, LIU H Y, et al. Analysis and Improvement of the Registration Method Based on SPM for Functional Magnetic Resonance Imaging [J]. Medical Imaging Engineering, 2015, 23

(10): 780-784. (in Chinese)

陈文, 郁芸, 刘宏毅, 等. 基于 SPM 的功能磁共振成像图像头动校正配准方法的分析与改进[J]. 医学影像工程学, 2015, 23(10): 780-784.

- [22] ZHANG Y C, FAN W, CHEN Z Q, et al. Application of SPM Technique in Time Stability of MRI System [J]. China Medical Equipment, 2014, 29(11): 19-22. (in Chinese)
- 张一驰, 樊伟, 陈自谦, 等. SPM 技术在磁共振系统时间稳定性方面的应用探讨[J]. 中国医疗设备, 2014, 29(11): 19-22.

(上接第 266 页)

有的 DPM 策略中进行动态选择; Shih 等人^[11]提出了一个自适应超时方案来适应突发服务到达模式的超时周期; Hwang 等人^[12]描述了一个基于收集和分析 I/O 设备访问模式信息的预测关闭方法。

使用概率模型检测技术进行 DPM 策略分析也逐渐被研究者所采用, Norman 等人^[13]提出了一个使用概率模型检测来分析随机 DPM 策略的方法, 并针对 IBM 磁盘驱动器的随机策略进行了验证。Cloth 等人^[14]研究了马尔科夫开销模型和带有状态开销的连续时间马尔科夫链, 并用此模型来验证 DPM 中的策略。

模型检测的合成技术已被用于部分领域, Chen 等人^[15]利用 PRISM-games 的合成功能对城市导航等系统合成策略, 并验证了其有效性。Cassez 等人^[16]使用 UPPAAL-TIGA 对一个油泵控制器合成策略, 为工业控制问题提供了一种新的解决思路。

结束语 本文通过对 DPM 策略生成与验证问题以及概率模型检测技术的研究, 提出了一个用概率模型检测技术对动态能耗管理中的策略进行合成与验证的方案。将动态能耗管理策略生成问题转换为概率模型检测中的策略生成问题, 并采用概率模型检测工具 PRISM-games 生成策略; 另外, 用模型检测工具 PRISM 对生成的 DPM 策略进行验证。实验表明, 该方案具有可行性和有效性。

参 考 文 献

- [1] BENINI L, BOGLIOLO A, PALEOLOGO A, et al. Policy optimization for dynamic power management[C]//Design Automation Conference, 1998. IEEE, 1999: 182-187.
- [2] NORMAN G, PARKER D, KWIATKOWSKA M, et al. Using probabilistic model checking for dynamic power management [J]. Formal Aspects of Computing, 2005, 17(2): 160-176.
- [3] KWIATKOWSKA M, PARKER D. Automated Verification and Strategy Synthesis for Probabilistic Systems[C]//International Symposium on Automated Technology for Verification and Analysis. Springer, Cham, 2013: 5-22.
- [4] CHEN T, FOREJT V, KWIATKOWSKA M, et al. PRISM-games: A Model Checker for Stochastic Multi-Player Games[C]//International Conference on TOOLS and Algorithms for the Construction and Analysis of Systems. 2013: 185-191.
- [5] KWIATKOWSKA M, NORMAN G, PARKER D. The PRISM Benchmark Suite[C]//Ninth International Conference on Quantitative Evaluation of Systems. IEEE Computer Society, 2012: 203-204.
- [6] KWIATKOWSKA M, PARKER D. Automated Verification and Strategy Synthesis for Probabilistic Systems[C]//International Symposium on Automated Technology for Verification and Analysis. Springer, Cham, 2013: 5-22.
- [7] Technical specifications of hard drive IBM Travelstar VP [DB/OL]. www.storage.ibm.com/storage/oem/data/travvp.htm.
- [8] KHAN, ALI U, RINNER, et al. Online learning of timeout policies for dynamic power management[J]. ACM Transactions on Embedded Computing Systems, 2014, 13(4): 96.
- [9] BENINI L, BOGLIOLO A, DE MICHELI G. A survey of design techniques for system-level dynamic power management [J]. Readings in Hardware/Software Co-Design, 2002, 8(3): 231-248.
- [10] DHIMAN G, ROSING T S. Dynamic power management using machine learning[C]//IEEE/ACM International Conference on Computer-Aided Design. IEEE, 2006: 747-754.
- [11] SHIH H C, WANG K. An adaptive hybrid dynamic power management algorithm for mobile devices[J]. Computer Networks, 2012, 56(2): 548-565.
- [12] HWANG Y S, KU S K, CHUNG K S. A predictive dynamic power management technique for embedded mobile devices[J]. IEEE Transactions on Consumer Electronics, 2010, 56(2): 713-719.
- [13] NORMAN G, PARKER D, KWIATKOWSKA M, et al. Using probabilistic model checking for dynamic power management [J]. Formal Aspects of Computing, 2005, 17(2): 160-176.
- [14] CLOTH L, KATOEN J P, KHATTRI M, et al. Model checking Markov reward models with impulse rewards[C]//International Conference on Dependable Systems and Networks, 2005 (DSN 2005). IEEE, 2005: 722-731.
- [15] CHEN T, FOREJT V, KWIATKOWSKA M, et al. Automatic verification of competitive stochastic systems[J]. Formal Methods in System Design, 2013, 43(1): 61-92.
- [16] CASSZE F, JESSEN J J, LARSEN K G, et al. Automatic Synthesis of Robust and Optimal Controllers—An Industrial Case Study[C]//International Conference on Hybrid Systems: Computation and Control. Springer-Verlag, 2009: 90-104.