

基于形式化模型的性能分析(FMPA)方法及工具设计概述

朱雪阳

中国科学院软件研究所计算机科学实验室

摘要

传统的基于测量的性能分析方法往往在代码实现后才考虑实施,而有些性能问题在开发过程的早期已经被引入,这时才予以考虑,为时太晚;因为修改必须回溯到系统的早期模型,势必增加开发成本及周期。因此,我们提出了一种基于形式化模型的性能分析(FMPA)方法,希望在系统开发的早期,利用模型对性能进行分析以尽早发现并解决性能问题。本文还概述了该方法支撑工具(FMPAer)的设计方案。

FMPA 方法旨在为实时嵌入式系统的设计开发提供多项性能指标的参考,适用于基于模型的开发过程。FMPA 具有统一对外接口(UML+MARTE)、基于多种形式化模型、可对多种系统性能指标进行分析的方法。FMPAer 的输入为 UML+MARTE 模型及所需分析的性能指标,输入为相关性能指标值。

1. 研究背景

随着计算机系统的广泛使用，软件性能问题得到越来越多的关注；特别是安全攸关的软件，其性能更是关键。在很多情况下，无法满足性能需求的软件已不能算正确。

传统的基于测量的性能分析方法在运行时对性能进行测试，往往在代码实现后才考虑实施。实践经验表明，有些性能问题既不能通过代码的性能调优加以解决，也不能通过增加硬件得到改善；这是由于这些问题是体系结构或设计因素造成的，而不是低效的编码引起的。这样的性能问题在开发过程的早期已经被引入，到集成测试或更晚的时候才予以考虑，为时太晚；因为修改必须回溯到系统的早期模型，势必增加开发成本及周期。

因此，有必要研究在系统开发的早期，利用模型对性能进行分析与优化方法以便尽早发现并解决性能问题，提高开发效率并降低成本。

统一建模语言 UML 作为面向对象建模语言，因其建模的灵活性和实用性，且可对系统的多个侧面进行建模，在软件开发中得到广泛应用，并已成为事实的工业标准。为了对实时嵌入式系统进行建模与分析，OMG 又提出 UML-MARTE (Modeling and Analysis of Real-Time Embedded system) 标准，对 UML 在性能和时间属性建模方面进行扩充。但 UML 缺乏精确的语义，难以直接对 UML 模型进行性能分析。形式化方法是基于数学的手段为软件和硬件系统提供精确规约、开发和验证的方法，以严格的形式化模型为中心，可以弥补 UML 在这方面的不足。但是形式化方法技术一般较难学习与掌握，因为在对系统进行性能分析时，需要软件设计人员对系统构造形式化模型，软件设计人员需要具有相关的数学知识背景。

如何结合 UML 模型与形式化方法的研究成果，形成优势互补，以对系统性能进行分析是一重要且复杂的科学问题。本文提出的 FMPA (Formal Models based Performance Analysis) 方法正是希望通过部分地解决这一问题，将形式化方法应用于软件性能分析，为实时嵌入式系统的开发提供多项性能指标评估参考。相关支撑工具 FMPAer (Formal Models based Performance Analyzer) 具有统一的对外接口(UML+MARTE)、基于多种形式化模型、可对多种系统性能指标进行分析。

2. FMPA 方法框架

FMPA 方法总体研究框架如图 1 所示。

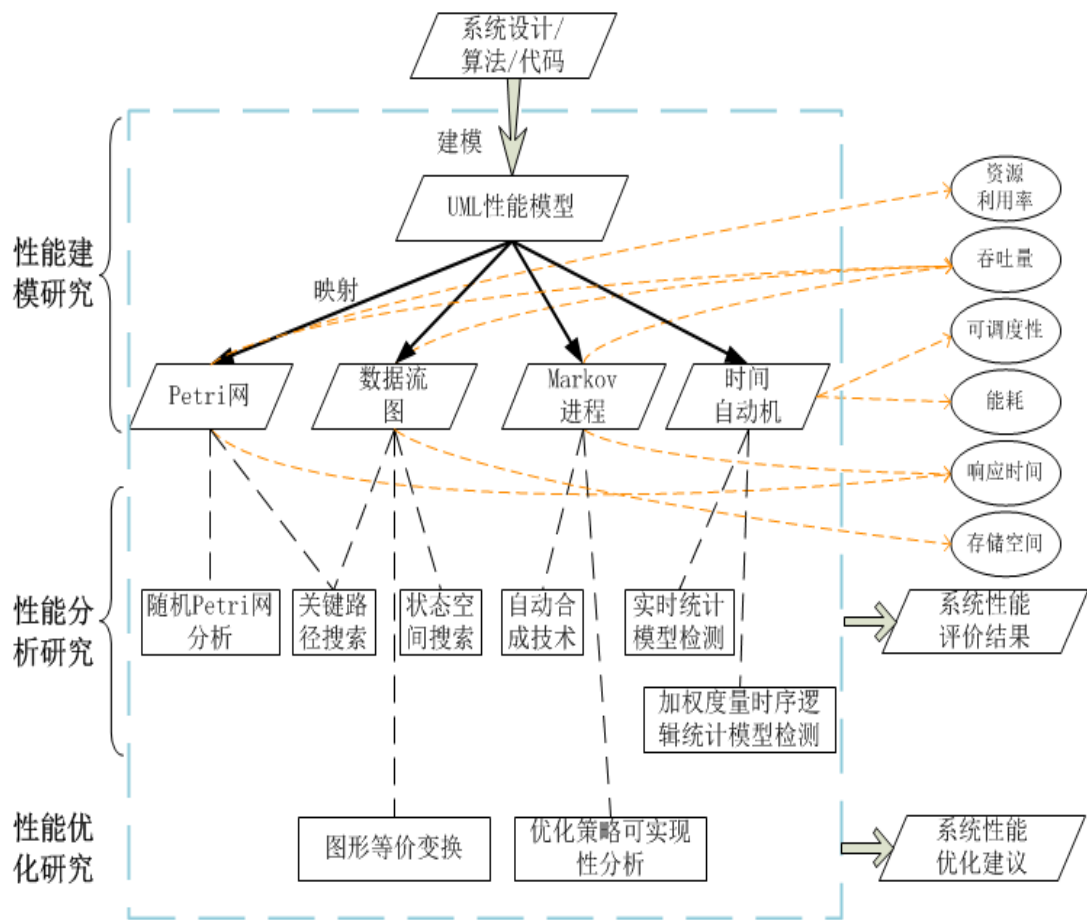


图1 FMPA 总体研究框架

以下就性能建模、性能分析及性能优化三方面简述之。

1) 性能建模

为了兼顾工程上使用的方便性及分析所需的严格性，FMPA 建模语言分抽象与具体两个层次：抽象层提供对外一致接口；具体层为形式化分析提供基础。为方便工程师使用，抽象层采用工业界的事实标准 UML-MARTE 中的性能模型；具体层为针对不同性能指标、不同系统特点，采用不同的具有严格语义模型的形式化语言，包括模型 Petri 网、数据流图、时间自动机及马尔柯夫决策过程等。UML-MARTE 用状态机、活动图、顺序图等行为图描述系统行为，相关性能分析参数作为注释，形成性能模型。针对不同的性能指标，我们将研究如何将这些行为图再度抽象为具体的形式化语言，只保留性能分析所需元素以提高分析效率。

双层建模语言不仅保证了对外接口的一致性及分析方法的多样性，还有足够的开放性以便于新分析模型的加入。加入新模型时，只要建立起 UML 性能模型到该模型的语义映射，即可将新的分析方法纳入我们的统一分析框架中。

2) 性能分析

这部分围绕上述四种形式化模型,研究利用不同技术手段对系统的资源利用率、可调度性、响应时间、吞吐量、能耗及存储空间等性能指标进行分析的方法。目前具体包括以下四类方法:基于模型检测的分析方法、基于随机 Petri 网的分析方法、基于数据流图的分析方法及基于合成的方法。

3) 性能优化

基于以上方法的分析结果,进一步提出系统性能优化的相关方法,为未来的系统设计提供参考。这部分包括以下两点:基于图形转换的方法、基于自动合成的方法。

总之, FMPA 是具有统一的对外接口、基于多种形式化模型、可对多种系统性能指标进行分析的方法。对于不同性能指标和不同系统的特点,不同的形式化方法在进行性能分析时各有优劣,利用多种形式化对系统进行分析可能得到更多性能指标的分析结果,为系统开发提供更有力的支持。

3. FMPAer 工具总体设计方案

FMPAer 工具结合多个自主开发与开源工具,实现从 UML 建模、模型转换、底层形式化方法性能分析、到返回分析结果的全过程。主要包括:将 UML 性能模型转换成各种形式化模型的语义转换、语义模型到各具体底层形式化分析工具语言的语法转换、相关性能指标自动转换为底层形式化分析工具的性质描述、将底层工具分析结果返回前端。以下简述 FMPAer 的总体设计及部署方案。

1) 开发环境及语言

Linux/Eclipse/Java/ XML

2) 总体设计框架

FMPA 使用的开源及自主开发工具主要包括:

- 关于模型与模型转换:
 - Papyrus (UML 图形建模工具, 开源)
 - ATL (M2M/Atlas Transformation Language, 开源)
- 关于底层形式化工具:
 - WebSPN (随机 Petri 网分析工具, 开源)
 - SPNP (随机 Petri 网分析工具, 开源)

- UPPAAL（实时模型检测工具，开源）
- CTAV（实时模型检测工具，自主研发）
- SDF3（数据流分析工具，开源）
- iDFOS（数据流优化与调度工具，自主研发）
- PRISM（概率模型检测工具，开源）

其中底层形式化工具，我们还将随着调研的深入进行，调整并吸收进更多工具以增强 FMPAer 的功能及效率。

FMPAer 的总体设计框架如图 2 所示。

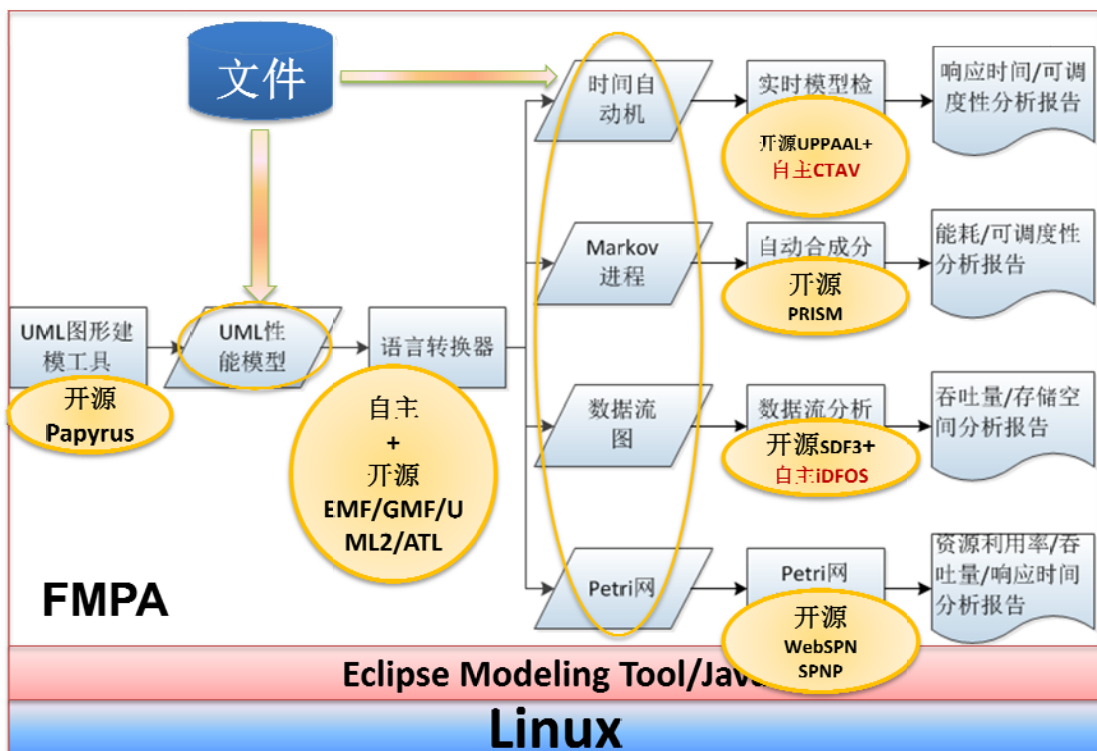


图2 FMPA 总体设计框架

总体而言，FMPAer 子工具包括三大类，图形建模工具、语言转换器及形式化分析工具。为保证整体框架的灵活性与开放性，各工具之间以文件形式交互。以下分述之。

FMPAer 的图形建模工具，主要采用开源软件 Papyrus，为用户提供系统 UML-MARTE 模型的图形化编辑环境。

FMPAer 的语言转换器，包括从 UML 到各形式化语言转换的多个工具，采用语义语法两层分离的转换。语义转换层，采用元模型转换技术，根据基于 ATL 模型转换语言、必要时适当扩充的方法，将 UML-MARTE 元模型转换为各种形

式化模型元模型；语法转换层，将以 FMPAer 格式存储的各种形式化模型文件，转换为底层相应形式化分析工具的入口文件。

3) 部署方案

FMPAer 工具采用 C/S 架构。客户端含工具的集成界面及建模与语言转换，部署于本地执行；服务端为形式化分析工具，可于本地执行也可部署于高性能计算服务器。FMPAer 工具部署如图 3 所示。

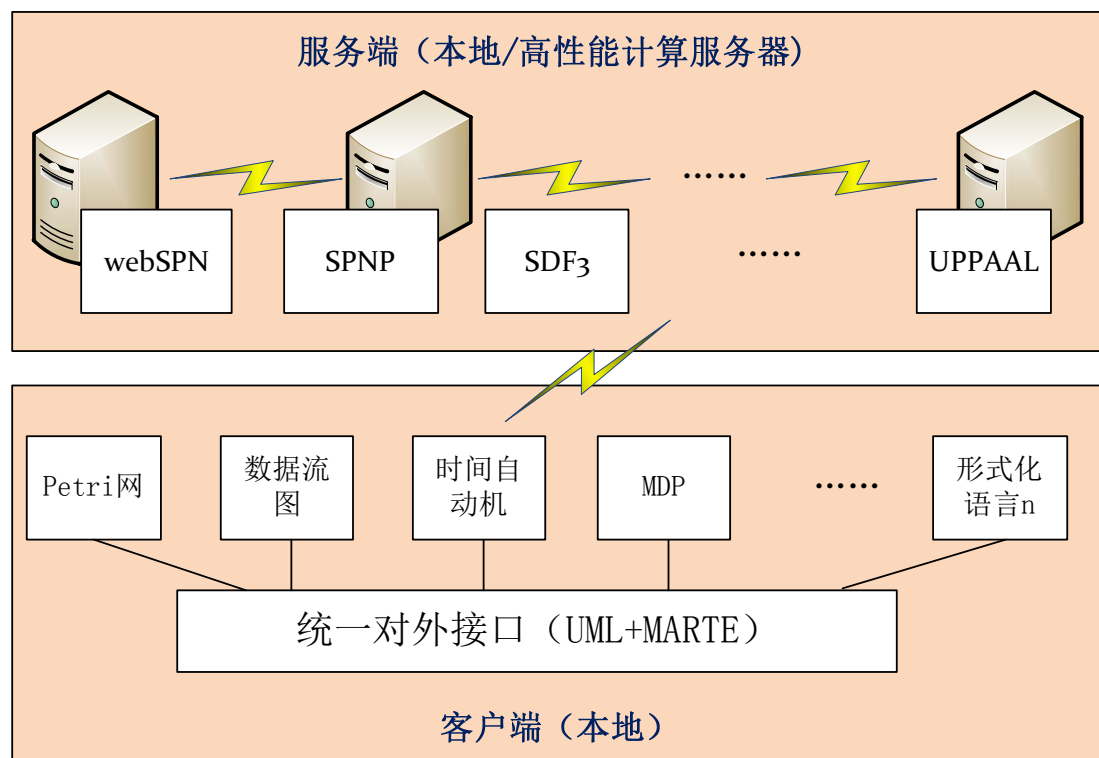


图3 FMPA 部署方案

4. 总结

本文针对传统的基于测量的性能分析方法的不足，提出了一种基于形式化模型的性能分析(FMPA)方法，希望在系统开发的早期，利用模型对性能进行分析以尽早发现并解决性能问题。FMPA 方法旨在为实时嵌入式系统的设计开发提供多项性能指标的参考，适用于基于模型的开发过程。FMPA 具有统一对外接口 (UML+MARTE)、基于多种形式化模型、可对多种系统性能指标进行分析的方法。

本文还概述了 FMPA 方法的支撑工具 FMPAer 的总体设计方案及部署方案，为下一步奠定基础。

缩写定义

FMPA (Formal Models based Performance Analysis): 基于形式化模型的分析方法

FMPAer (Formal Models based Performance Analyzer): FMPA 支撑工具

OMG (Object Management Group): 国际对象管理组织

UML (Unified Modeling Language): 统一建模语言

MARTE (Modeling and Analysis of Real-Time Embedded system)

UML-MARTE: UML Profile for MARTE

ATL (M2M/Atlas Transformation Language): 开源模型转换语言

SPN (Stochastic Petri Nets): 随机 Petri 网

SDF (Synchronous Data Flow): 同步数据流

MC (Markov Chain): 马尔科夫链

MDP (Markov Decision Process): 马尔科夫决策进程

iDFOS (DataFlow Optimization and Scheduling Tool): 自主研发的数据流优化与调度工具

SDF3: 开源数据流分析工具

CTAV: 自主研发的实时模型检测工具

UPPAAL: 开源实时模型检测工具

WebSPN (Web-accessible non Markovian Petri net tool): 开源随机 Petri 网分析工具

SPNP (Stochastic Petri Net Package): 开源随机 Petri 网分析工具

PRISM: 开源概率模型检查工具(probabilistic model checker)。PRISM 可以分析的概率模型有：离散时间马尔科夫链 (DTMC)，连续时间马尔科夫链 (CTMC)，马尔科夫决策过程 (MDP)，概率自动机 (PAs) 和概率时间自动机 (PTAs)。