

多目标约束下软件运行时验证加速技术框架

刘彦斌¹, 王毅刚², 叶飞²

(1. 中国电子科技集团公司第十三研究所, 石家庄 050051;

2. 中国人民解放军军械工程学院, 石家庄 050003)

摘要: 软件运行时验证是一种近年来逐步兴起的通过监控程序运行来检验其是否满足给定性质的轻量级验证技术。由于复杂性质的运行时验证中常产生高额的时间开销, 阻碍了该技术在部署后系统中的应用。在深入剖析国内外研究现状及存在问题基础上, 从改善部署后软件运行时验证效率的角度出发, 综合考虑性质违背检测能力、诊断支持能力等潜在开销优化制约因素, 提出多目标约束下的软件运行时验证加速技术框架。该框架包括构建多目标约束模型、可加速监控器判定、加速控制技术研究和原型工具开发等内容, 并具体阐述了框架所涉及的关键技术方案。本研究将为解决运行时验证中的开销问题提供关键技术支撑, 为运行时验证技术在部署后系统中的工程化应用奠定基础。

关键词: 运行时验证; 多目标约束; 监控器; 监控开销; 运行时监控

本文引用格式: 刘彦斌, 王毅刚, 叶飞. 多目标约束下软件运行时验证加速技术框架[J]. 兵器装备工程学报, 2016(8): 88-92.

Citation format: LIU Yan-bin, WANG Yi-gang, YE Fei. Runtime Verification Speeding up Based on Multi-Objective Constraint[J]. Journal of Ordnance Equipment Engineering, 2016(8): 88-92.

中图分类号: TP311

文献标识码: A

文章编号: 2096-2304(2016)08-0088-06

Runtime Verification Speeding up Based on Multi-Objective Constraint

LIU Yan-bin¹, WANG Yi-gang², YE Fei²

(1. The 13rd Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050051, China; 2. Ordnance Engineering College of PLA, Shijiazhuang 050003, China)

Abstract: Runtime verification is a relatively new lightweight formal verification method which is concerned with dynamic monitoring and analysis of system executions with respect to precisely specified properties. Runtime verification of complicated properties that involve many system variables imposes high overhead, which makes it is hard to apply this method in the context of deployed systems. This paper considered all the latent constraints including the efficiency of property violation detection and fault diagnosis. It involved the key technology at the topic of speeding up for runtime verification on the condition of multi-objective constraint. It mainly includes: to construct a model of multi-objective constraint, to resolve the problem of how to recognize the involved monitor for speeding up by constructing heuristic algorithm to get the answer for multi-objective model on the basis of monitoring statistics information of verified systems before speeding up, using control theory, to implement runtime verification speeding up by constructing self-adaptive speeding up control algorithm. This research will provide key technologies for resolving the overhead problem of runtime verification. And it can also provide a basis for

收稿日期: 2016-02-29; 修回日期: 2016-03-26

基金项目: 河北省自然科学基金项目“多目标约束下的软件运行时验证加速技术研究”(F2014506017)

作者简介: 刘彦斌(1978—), 男, 博士, 高级工程师, 主要从事可信软件研究; 王毅刚(1975—), 男, 博士; 叶飞(1979—), 男, 博士。

applying the runtime verification method for actual deployed systems.

Key words: runtime verification; multi-objective constraint; monitor; monitoring overhead; runtime monitoring

运行时验证(Runtime Verification)是一种近10多年来逐步兴起的针对程序具体运行的轻量级验证技术。它把形式化验证技术和系统的实际运行结合起来,通过监控程序运行并检验其是否满足给定性质实现对系统的验证。典型的实现方式是:在静态阶段,根据性质规约转换生成监控器(Monitor),并插桩程序使之能够在运行中向监控器发送相关事件;在动态运行阶段,监控器处理这些事件序列并决定是否满足性质规约。由于运行时验证针对的对象仅仅是系统运行中的单个或者少数执行轨迹,不需要针对整个系统模型,从而避免了状态空间爆炸问题。而且,监控器能够被集成成为被验证系统的一部分,不但可以用来检测软件运行中的错误,也可在检测到性质违背时及时采取措施(如,执行修复代码等),从而提供额外的系统运行阶段安全保障。

伴随软件运行时验证技术的发展,在常规的状态性质、时序性质,以及并发程序涉及的并发性质^[17]之外,人们希望通过运行时验证技术来验证更为复杂的系统属性。比如,参数化性质,这类性质往往要求特定类型的所有对象或数据结构的所有实例都满足或不满足。在面向对象程序中,经常会要求其动态实例化后产生的所有对象都满足某个特定时序性质;在并发程序中,要求所有动态创建的进程都满足某个性质^[15]。对参数化性质等复杂性质的运行时验证也是当前的研究热点。

但是,参数化性质等较为复杂性质的在线运行时验证中,由于需要维持大量的监控器实例并处理程序实体中生成的大量事件,经常产生高额的内开销(Overhead)。研究表明,对于当前具有代表性的运行时验证工具JavaMOP^[9]和Tracematches^[8],所产生的平均开销分别为41%和112%,一些复杂性质产生的开销能达到程序执行时间的970%^[18],如此高的开销,是用户所不能容忍的。尤其对于部署后的实际用户使用环境中的程序而言,用户所能容忍的开销更低。根据Bodden等^[8]的调查结果,工业界普遍所能接受的开销应当在5%以内。开销问题阻碍了运行时验证技术在实践中被广泛用于软件部署后环境,而大部分仅仅停留在软件测试过程中使用。如何减小软件运行时验证开销,已成为当前亟待解决的难点问题。

本文在深入剖析国内外研究现状及存在问题基础上,从改善部署后软件运行时验证效率的角度出发,综合考虑性质违背检测能力、诊断支持能力等潜在开销优化制约因素,提出多目标约束下的软件运行时验证加速技术框架。通过开展多目标约束模型构建、可加速监控器判定、加速控制等关键技术研究以及相关支持工具研制,实现运行时验证加速目的。该研究将为解决运行时验证中的开销问题提供关键技术支撑,为运行时验证技术在部署后系统中的工程化应用奠定基础。

1 国内外研究现状

近年来,软件运行时验证技术在国际上备受关注,包括美国国家航空航天局 AMES 研究中心、伊利诺伊大学香槟分校、宾夕法尼亚大学、慕尼黑工业大学、IBM Haifa 研究中心等在内的许多国际研究机构正持续开展运行时验证理论与方法研究。研究方向包括性质规约与在线检验算法^[2-4]、轨迹生成方法^[5]、运行时执行(Runtime Enforcement)与反馈^[6]等。开发的代表性的运行时验证工具包括:JavaMaC^[7]、Tracematche^s^[8]、MOP^[9-10]、RuleR^[11]、JPaX^[12]、QVM^[13]等。国内,国防科技大学、北京大学等单位也开展了相应的研究。例如,董威等^[14-16]研究了主动监控理论及预测监控器的生成方法。针对软件运行时验证加速的研究主要包括以下几类:

1) 联合静态分析减小开销。通过静态分析可减少从程序中提取事件所需的插桩点数量。Dwyer等^[19]利用静态分析来移除Typestate性质监控中不必要的插桩。Bodden等^[20-22]探索了许多轻量级的代码静态分析方法,通过识别可安全移除的插桩点减小开销。Purandare等^[23]优化了程序中性质相关的循环结构,通过仅仅监控有限次数的循环减小开销。

2) 基于取样的开销优化方法。该类技术通过对性质或者事件取样减小开销。Diep等^[24]提出了基于格的取样技术减小路径性质监控中的开销。Bonakdarpour等^[25]提出了时间触发的运行时验证概念,监控器周期性地取样程序状态评估性质是否成立。Navabpour等^[26]研究了启发式算法求解给定取样周期内需要被缓冲的最小数量的关键事件,以便时间-触发的监控器能够成功重构两个连续取样时间内的程序状态。Goodloe等^[27]从语言层面提供了基于取样的监控机制。Arnold等^[13]构建了被称为质量虚拟机(QVM)的特殊运行时环境,采用了性质指导的取样和对象为中心的取样策略。

3) 基于多用户协同的开销分解方法。Bodden等^[8]提出了多用户协同的运行时验证技术。该技术通过将运行时验证任务分配给多个用户,每个用户仅仅执行部分插桩的程序,从而减小每个用户所承担的开销。该技术源自Liblit等^[28]所提出的基于统计方法的协同bug诊断思想,利用源自多用户的大量程序执行轨迹的信息组合进行失效的隔离。

通过直接或者间接减小插桩点和监控器数量减小开销,是运行时验证中开销优化的基本思路,如图1所示。虽然现有研究在开销优化方面取得了一定的成果,可以有效减小插桩点数量(如联合静态分析、事件取样)和减小监控器数量(如性质取样)。但当前研究尚存在许多不足之处:首先,已

有研究没有充分考虑开销优化中的潜在制约因素,没有充分考虑开销优化对诊断支持能力的潜在影响,在优化过程中可能会损害系统正常功能。其次,难以满足工程界的“有界开销”需求。当前研究主要集中于“尽可能减小开销”,从技术本质而言,所产生的开销仍然是“无界的(Unbounded)”。而目前工程中更关注的是如何使得在各种条件下开销都能满足要求,即“有界(Bounded)”的开销。尽管有学者^[29]初步探索了控制方法产生有界开销,但适用范围尚很有限。

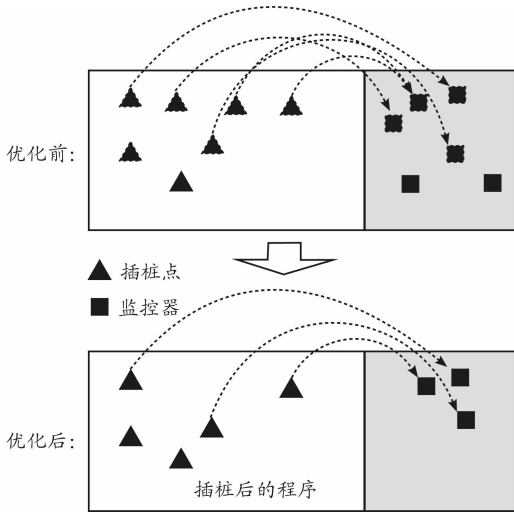


图1 开销优化示意图

2 运行时验证加速技术框架

针对当前研究中存在的上述问题,本研究从改善部署后软件运行时验证效率的角度出发,不仅将开销作为优化目标,还同时考虑开销优化对验证结果准确性以及诊断支持能力的影响。把软件运行时验证的优化归结为多目标优化问题,以减小监控开销、提高验证精度、改善诊断支持能力为优化目标,构建了多目标约束下的软件运行时验证加速技术框架。所构建的框架如图2所示,主要包括软件运行时验证中的多目标约束建模、可加速监控器判定技术、加速控制技术、工具原型开发及应用试验等内容。

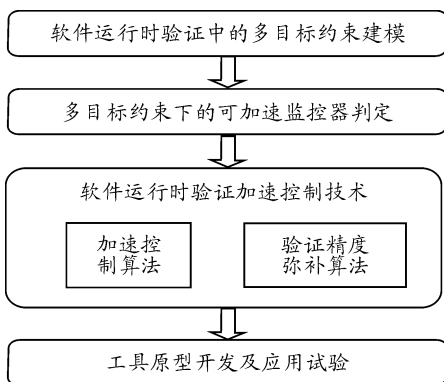


图2 技术框架示意图

1) 软件运行时验证中的多目标约束建模

在软件运行时验证中,由于性质验证和诊断所需信息源常常重叠,优化监控开销不仅影响性质违背检测能力,而且能够影响诊断所需信息的收集。软件运行时验证加速过程需要在减小监控开销、提高验证精度和改善诊断支持能力多个互相制约的目标间进行权衡。

该技术以实验分析方法为基础,识别运行时验证优化过程中多目标间的内在依赖关系,抽取出细粒度的影响变量,量化多目标约束,构造开销量化模型、性质违背检测能力评估模型和诊断支持能力评估模型,进而建立多目标约束模型,作为软件运行时验证加速的分析依据。

2) 多目标约束下的可加速监控器判定

在实施运行时验证加速之前,需要首先判定出哪些监控器是“可加速”的,即判定出哪些监控器在运行中不再必要、哪些可以进行动态调整,且满足多目标约束。对于参数化性质等复杂性质而言,由于监控器数量庞大,存在动态性、不确定性等特征,且监控开销、验证精度和诊断支持能力之间常常发生冲突,满足多目标约束的可加速监控器的判定是非常困难的。

该技术将探索基于多目标模型求解的可加速监控器判定技术。利用性质的语义结构分析和从程序运行中获得的执行轨迹等监控信息,研究启发式算法进行监控器“可加速”判定。将利用程序执行轨迹等监控信息,一方面结合统计分析和机器学习技术,构建被验证系统的统计模型,挖掘不同监控器之间的潜在联系,判定某监控器是否必要;另一方面用来迭代确定多目标模型中的相关参数值。该技术能够识别出满足多目标约束的可加速监控器以及相应的加速操作方式,作为实施软件运行时验证加速的输入。

3) 软件运行时验证的加速控制技术

为满足工程界对“有界(Bounded)开销”的需求,该技术将研究运行时验证的加速控制技术,利用反馈控制机理,动态调整监控器,将开销控制在用户可接受范围之内;同时,在加速过程中,必须满足多目标约束,不能影响验证精度。由于传统的比例-积分-微分(PID)控制、监督控制等理论不能直接应用在线性离散系统中,将针对被验证系统的特点,研究相应的加速控制实施机制和算法。所研究的自适应加速控制算法,能够在软件运行过程中,以可加速监控器及相应的加速操作方式为输入,利用系统当前实际开销和目标阈值间的误差进行反馈控制,自适应地对监控器进行增量式在线动态调整(例如,临时关闭某监控器),避免监控开销超出用户所能接受的水平。

4) 在上述研究基础上,研制支持工具原型并进行应用试验

基于以上研究成果,设计实现相应的支持工具原型,支持多目标约束模型构建,可加速监控器判定以及加速控制技术的实现,并在具体案例中进行应用试验。进而,根据实际效果和应用反馈进一步改进所研究的模型与方法。

3 多目标约束模型

本文将监控器标识为 m , 所对应的监控器集合 $m = (m_1, m_2, \dots, m_n)^T$ 是 N 维向量, m 所在的空间是决策空间。 m 对应的目标函数分别标识为 $f_1(m), f_2(m)$ 和 $f_3(m)$, 其中: $f_1(m)$ 为性质违背检测能力评估函数; $f_2(m)$ 为监控开销量化评估函数; $f_3(m)$ 为诊断支持能力评估函数。

三维向量 $(f_1(m), f_2(m)$ 和 $f_3(m))$ 所在空间是目标空间。

可加速监控器的判定: 对监控器 $M = (m_1, m_2, \dots, m_n)$ 中的可加速监控器进行判定的过程, 就是求解下列约束方程 $Cmul$ 得到监控器的过程:

$$Cmul = \begin{cases} f_1(M) < f_1(M') \\ f_2(M) > f_2(M') \\ f_3(M) \leq f_3(M') \end{cases} \quad (1)$$

$Cmul$ 就是本文所构建的多目标约束模型, 它的具体推导过程将在别的文献中具体阐述。利用性质的语义结构分析和从程序运行中获得的执行轨迹等监控信息, 已构建了启发式算法对该多目标约束方程 $Cmul$ 进行求解。

4 实验与验证

针对上述技术框架, 采用某卫星控制系统作为案例开展实验。该卫星载有多种执行各种任务的设备(如照相机、温度传感器等), 地面人员通过操作指令可对卫星进行控制。

卫星上发生的每个重要事件都被记录在日志中并传回给地面, 地面日志模块接受并存储这些事件, 将通过这些数据对软件进行运行时验证。

所开展实验的具体步骤及技术路线如图 3 所示。主要包括数据准备、多目标约束模型构建、程序执行轨迹统计分析、可加速监控器判定、加速控制算法和验证精度补偿等主要步骤。

1) 数据准备。首先, 采用受控实验方法获取某卫星模型控制系统运行时验证相关实验数据。选取了 Command-Success、SAFEFILEWRITE 等各类不同类型的性质作为待验证性质。例如, 该卫星系统期望行为应满足 CommandSuccess 性质: 每个 Command (i, n, t_1) 事件应当最终跟随 Success (i, n, t_2) 事件, 在期间不能有 Fail (i, n, t_3) 事件发生。该命令成功 (CS, command success) 性质可以用 LTL 进行表达。在实验环境下, 收集其加速前完整的程序执行轨迹信息, 共生成 500 个轨迹, 每个轨迹包含 400 个命令, 平均轨迹长度是 2 000 个事件。此外, 还收集了监控开销、验证精度、诊断支持能力相关的实验数据。

2) 多目标约束模型构建及求解。根据所收集实验数据, 结合定性评估和定量分析方法, 建立初步的开销度量函数、性质违背检测能力评价函数和诊断支持能力评价函数, 通过考察三者之间的内在依赖关系, 得到多目标约束模型 $Cmul$ 的具体参数。由于各个目标之间相互存在冲突, 采用了启发式迭代算法对多目标模型求解, 得到决策空间上的最优解 M' , 并将 M' 中的监控器转换插入目标软件, 从而对其进行运行时监控。

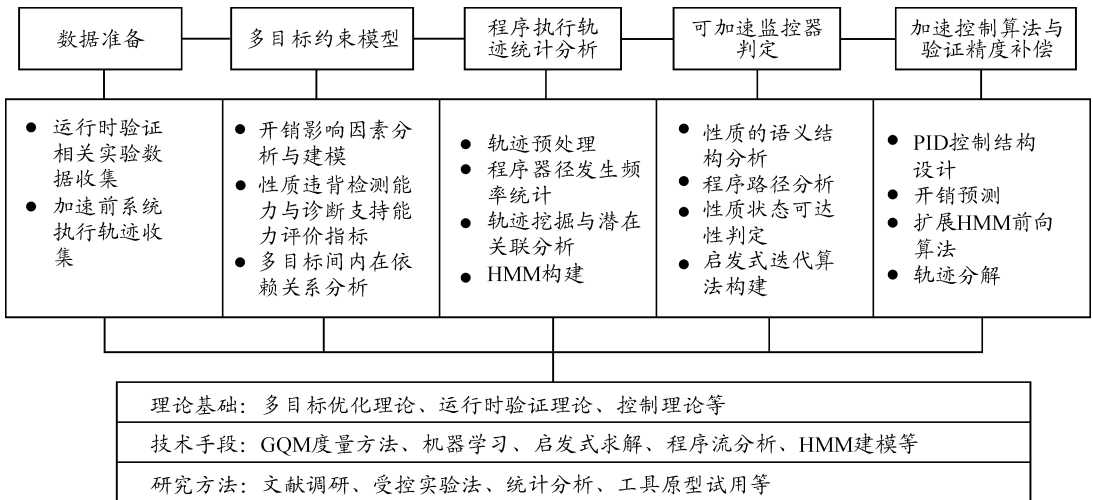


图 3 实验步骤及技术路线示意图

3) 结合统计分析和机器学习方法, 对目标系统运行中获得的执行轨迹等监控信息进行统计分析, 构建目标系统的统计模型。由于在隐马尔可夫模型中, 根据可观察状态能够确定系统的隐含状态。从系统完整的执行轨迹中学习并构建了隐马尔可夫统计模型 (HMM), 并利用该模型来弥补加速过程中可能造成的监控事件缺失。

4) 应用控制理论, 将加速控制问题转化为对非线性系统设计一个最优控制器的问题。改进并联合面向离散时间系统的 PID 控制器和面向离散事件系统的监督控制理论, 设计满足约束的控制器。在此基础上, 设计了自适应加速控制算法, 使之能够根据开销量化模型进行开销预测, 实现费用感知的加速控制。

通过对比加速前后的实验数据,结果表明,在加速前平均开销为 51%,加速之后在满足验证精度、诊断支持能力前提下平均开销减小为 28%,通过加速控制算法临时关闭某些监控器可将平均开销控制在 10% 以内。

5 结束语

通过深入剖析国内外研究现状及存在问题,本文提出多目标约束下的软件运行时验证加速技术框架。与现有运行时验证开销优化方法相比,其特色和创新之处主要在于:①从多目标优化角度研究软件运行时验证的加速技术,通过建立多目标优化模型优化调整“监控器”,提高系统性能;②融合控制理论、机器学习等多学科技术,实现软件运行时验证加速。初步实验结果表明,本文所提方法能够在满足多目标约束前提下有效减小验证开销,实现验证加速目的。下一步将在本文研究基础上,进一步开展对比实验改进框架中的各项关键技术。

参考文献:

[1] 张献,董威,齐治昌.基于 AOP 的运行时验证中的冲突检测[J].软件学报,2011(6).

[2] BAUER A, LEUCKER M, SCHALLHART C. The good, the bad, and the ugly—but how ugly is ugly? [C]//Proceedings of the 7th International Workshop on Runtime Verification (RV'07). 2007:126–138.

[3] BAUER A, LEUCKER M, SCHALLHART C. Comparing LTL Semantics for Runtime Verification[J]. Journal of Logic and Computation, 2010, 20(3):651-674.

[4] MEREDITH P O. Efficient monitoring of parametric context-free patterns [J]. Automated Software Engineering, 2010. 17(2):149-180.

[5] SEYSTER J, DIXIT K, HUANG X, et al. Aspect-oriented instrumentation with GCC [C]//Proceedings of the 1st International Conference on Runtime Verification. LNCS, 2010:405–420.

[6] FALCONE Y, FEMANDEZ J C, MOUNIER L. What can you verify and enforce at runtime? [Z]. Software Tools for Technology Transfer, Special Section on Runtime Verification, 2011.

[7] KIM M. Java-MaC: A run-time assurance approach for Java programs [J]. Formal Methods in System Design, 2004, 24(2):129-155.

[8] BODDEN E. Collaborative Runtime Verification with Tracematches [J]. Journal of Logic and Computation, 2010, 20(3):707-723.

[9] CHEN F, GOSU G. MOP: An efficient and generic runtime verification framework [J]. Acm Sigplan Notices, 2007, 42

(10):569-588.

[10] MEREDITH P, JIN D, GRIFFITH D, et al. An overview of the MOP runtime verification framework [Z]. Software Tools for Technology Transfer. Special Section on Runtime Verification, 2011.

[11] BARRINGER H, RYDEHEARD D, HAVELUND K. Rule Systems for Run-time Monitoring: from EAGLE to RULER [J]. Journal of Logic and Computation, 2010, 20(3):675-706.

[12] HAVELUND K, ROSU G. An overview of the runtime verification tool Java PathExplorer [J]. Formal Methods in System Design, 2004, 24(2):189-215.

[13] ARNOLD M, VECHEV M, YAHAV E. QVM: An Efficient Runtime for Detecting Defects in Deployed Systems [J]. ACM Trans Softw Eng Methodol, 2011, 21(1):1-35.

[14] WEI D, MARTIN L, SCHALLHART C. Impartial anticipation in runtime verification [C]//Proceedings of the 6th International Symposium on Automated Technology for Verification and Analysis, Springer-Verlag, 2008.

[15] 赵常智,董威,隋平,等.面向参数化 LTL 的预测监控器构造技术[J].软件学报,2010(2):318-334.

[16] ZHAO C Z, JI W D, SUI P, et al. Software Active Online Monitoring Under Anticipatory Semantics [C]//1st International Workshop on Software Health Management 2009: Pasadena, California, USA.

[17] QADEER S, TASIRAN S. Runtime verification of concurrency-specific correctness criteria [Z]. Software Tools for Technology Transfer, Special Section on Runtime Verification, 2011.

[18] BODDEN E, HENDREN L, LHOTAK O. A staged static program analysis to improve the performance of runtime monitoring [C]//European Conference on Object-Oriented Programming, 2007:525–549.

[19] DWYER M, PURANDARE R. Residual dynamic typestate analysis. In: Int'l [C]//Conf on Aut Soft Eng, 2007:124–133.

[20] ERIC B, LAURIE H. A staged static program analysis to improve the performance of runtime monitoring [Z]. 2007.

[21] BODDEN E. Efficient hybrid typestate analysis by determining continuation-equivalent states [C]//Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering, ACM: Cape Town, South Africa. 2010.

[22] BODDEN E, LAM P, HENDREN L. Finding programming errors earlier by evaluating runtime monitors ahead-of-time [C]//Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, ACM: Atlanta, Georgia. 2008:36-47.

4 结束语

大数据时代的到来,预示着一场新的军事革命的即将来临,伴随部队信息化建设的发展,信息化管理平台在各部门的广泛使用,使得数据信息成为部队战斗力生成模式转变的新型催化剂、融合剂。由于数据的多结构,数据量的剧增,仅靠传统的数据分析,必然造成数据利用的局限性,时效性差,为了更好地利用部队数据,实现数据由辅助决策到支持决策,真正做到“数据到决策”的转变,必须寻求更适于多源异构的大数据分析手段,因此大数据分析技术成为部队未来的必然选择。同时在面临我军体制编制大变革的关键时期,构建适用于我军发展的大数据系统平台,对部队信息化的发展也具有强大的推动作用。

参考文献:

- [1] 杨巨龙. 大数据技术全解基础、设计、开发与实践[M]. 北京:电子工业出版社,2014.
- [2] 赵刚. 大数据技术与应用实践指南[M]. 北京:电子工业出版社,2014.
- [3] 李军. 大数据从海量到精准[M]. 北京:清华大学出版社,2015.
- [4] 邢程,李玉梅,胡荣亮.“大数据”下的军队信息资源建设[J]. 教育技术研究,2014(3):65-68.
- [5] 姜念云,张松海,谢夏. 大数据分析技术在文化资源管理中的应用[J]. 中国基础科学,2014(1):17-20,27.
- [6] 王向博. 基于数据挖掘的装备维修信息管理研究[D]. 军械工程学院装备指挥与管理系,2013.
- [7] 涂新莉,刘波,林伟伟. 大数据研究综述[J]. 计算机应用研究,2014,31(6):1612-1616,1623.

(责任编辑 杨继森)

(上接第92页)

- [23] PURANDARE R, DWYER M B, ELBAUM S. Monitor optimization via stutter-equivalent loop transformation [C]//Proceedings of the ACM international conference on Object oriented programming systems languages and applications, ACM:Reno/Tahoe, Nevada, USA. 2010:270-285.
- [24] DIEP M M, DWYER M B, ELBAUM S. Lattice-Based Sampling for Path Property Monitoring [J]. ACM Trans Softw Eng Methodol, 2011, 21(1):1-43.
- [25] BONAKDARPOUR B, NAVABPOUR S, FISCHMEISTER S. Sampling-based runtime verification [C]//Proceedings of the 17th international conference on Formal methods, Springer-Verlag: Limerick, Ireland. 2011:88-102.
- [26] NAVABPOUR S, WU W W C, BONAKDARPOUR B, et al. Efficient techniques for near-optimal instrumentation in time-triggered runtime verification [C]//Runtime Verification, 2011.
- [27] PIKE L. Copilot: a hard real-time runtime monitor [C]//Proceedings of the First international conference on Runtime verification, Springer-Verlag: St. Julians, Malta. 2010:345-359.
- [28] JIN G. Instrumentation and sampling strategies for cooperative concurrency bug isolation [C]//Proceedings of the ACM international conference on Object oriented programming systems languages and applications, Reno/Tahoe, Nevada, USA. 2010:241-255.
- [29] CALLANAN S. Software monitoring with bounded overhead [C]//Parallel and Distributed Processing, IPDPS 2008.
- [30] DWYER M B, PURANDARE R, PERSON S. Runtime verification in context: can optimizing error detection improve fault diagnosis [C]//Proceedings of the First international conference on Runtime verification, Springer-Verlag: St. Julians, Malta. 2010:36-50.
- [31] JIN D. Garbage collection for monitoring parametric properties [C]//Proceedings of the 32nd ACM SIGPLAN conference on Programming language design and implementation. San Jose, California, USA. 2011:415-424.

(责任编辑 杨继森)