“可执行文件”数据挖掘初探

——利用数据挖掘技术探求计算机软件领域所面临问题的解决途径

吴章金*

兰州大学信息科学与工程学院软件与理论专业

摘要

随着计算机科学与技术的发展，计算机软件领域面临着越来越复杂的问题，不过也积累了大量的相关数据和技术，本文试图利用数据挖掘技术，结合相关的软件技术，以“可执行文件”作为突破口对这些相关的数据进行分析和处理，找出可用的模式以便找出计算机软件领域所面临问题的解决途径。

关键字： 数据挖掘；可执行文件；软件技术；软件安全；软件运行效率；软件移植

1. 引言

随着计算机科学与技术的飞速发展，人类的生活和工作方式正在发生翻天腹地的变化。但是技术是一般双刃剑，在给人类带来“福利”的同时，也隐藏着潜在的安全危机，比如计算机病毒的传播和感染造成了极大的财产甚至生命的损失；与此同时，随着新的应用领域的不断拓展和涌现，对计算机技术本身提出了更多的需求，包括降低研发和设备投入成本、提高人员和机器的工作效率等。因为计算机软件是计算机完成各种工作的载体，因此，这其中比较重要和突出的问题是计算机软件领域所面临的问题，比如软件安全，软件运行效率，软件开发成本，软件可移植性等。

本文试图从计算机软件的最终载体和形式——“可执行文件”上找到突破口，通过计算机科学与技术本身的一个分支——数据挖掘技术来寻求解决相关问题的潜在途径，进而确实找到降低软件开发成本，提高软件运行效率，提高软件安全性和可移植性的可能方式。

下面将从这么几个方面来做介绍，先简要介绍计算机软件领域所面临的一些问题，接着寻找解决相关问题的突破口，引出对“可执行文件”的深入介绍，然后对引入数据挖掘技术的可行性进行分析，最后介绍一个相关的实例。

2. 软件领域所面临的问题

计算机软件领域涉及很广泛，也潜在着各种各样的问题，我们这里仅列举几个问题作为展开后续讨论的起点。

- 如何确保开发出的软件确实能够满足客户的需求，它涉及到哪些因素，如何改进？
- 如何降低软件开发的成本，软件开发成本涉及到哪些方面，如何降低？
- 如何提高软件的运行效率，除了算法外，还涉及什么,如何最大限度的节约时间，提高效率？
- 如何节约软件的存储空间，以便软件能够不受限制地在更多的领域应用，比如拓展软件在嵌入式领域的应用？可以通过改进软件的设计架构还是可以通过设计更好的可执行文件格式来实现呢？
- 如何提高软件的安全性，如何保障软件更不容易被病毒感染，如何降低软件本身的破坏性，仅仅是提高杀毒技术吗？还是从软件本身寻找原因？

解决这些问题必须寻找到一个合适的突破口，下面我们将试图寻求这样的突破口。

3. 寻求解决问题的突破口

本节将从“可执行文件”的定义、“可执行”文件的属性，以及可执行文件和软件领域所面临问题之间的关系三个层面切入来介绍为什么我们可以把“可执行文件”作为解决软件领域所面临问题的突破口。

3.1 什么是可执行文件

这里是wikipedia.org的定义：

“An executable or executable file, in computer science, is a file whose contents are meant to be treated as a program by a computer.”[1]

它从计算机科学的角度给予了定义，“可执行文件”是指一种其内容被计算机视为程序的文件。

那么它和软件的其他程序文件有什么区别呢？比如程序源代码？可重定向的目标代码？

另外一个来自engineering.purdue.edu的定义给我们澄清了这种区别。

* wuzhangjin@gmail.com

“a file that may be executed by typing its name as a command.”[2]

该定义指出可执行文件是一种在我们作为命令键入后可以被执行的一种文件。这就是它和其他程序文件的区别的。

从这样两个定义出发，我们可以这么说“可执行文件”是计算机软件实现某种功能的最终载体和形式。下面我们将分析它的各种属性，包括功能，开发效率、大小（占用的字节数）、运行平台、运行效率等等对计算机软件本身的质量、成本、应用场合等诸多方面的影响。

3.2 可执行文件的属性

可执行文件作为一个普通的文件它有如下的属性：文件类型，大小；而作为一个可执行文件，它可能有本身特定的结构（格式），执行权限，开发效率，功能，运行效率，支持的操作系统平台，安全性（是否容易注入病毒）等诸多属性。

就文件类型而言，有纯文本的脚本文件，也有二进制的目标程序文件。

就大小而言，有的可能几个字节，有的可能上兆字节。

就本身格式而言，因为脚本文件除了遵循特定的脚本语言语法外，没有额外的结构限制，所以这里仅考虑目标程序文件，比如早期 unix 平台上的 a.out 和 coff 格式，现代操作系统平台上普遍支持的 ELF 文件格式。

就执行权限而言，有些平台下根本没有这方面的考虑，而有些平台下引入了一个可执行权限位，用于告知操作系统是否可以运行一个可执行文件。

就功能而言，不同的可执行文件基本上都为了特定的应用目标而开发。

就运行效率而言，包括算法、处理器、内存等内在和外在的因素都可能对此造成很大的影响。

就支持的操作系统平台而言，有的仅支持某一个特定的平台；而有的则在很多操作系统平台下都支持，有更好的可移植性。

就安全性而言，有的可执行文件有天然的设计缺陷，存在很多“空洞”（例如全零的区域）[3]，容易注入病毒。

3.3 可执行文件和软件领域所面临的问题的关系

结合“可执行文件”的上述属性，我们不难发现，它们和计算机软件领域所面临的诸多问题存在密切的关系。

例如，“可执行文件”支持的运行平台情况和软件本身的可移植性关系很大，能够直接影响软件开发和生产的成本；“可执行文件”本身设计的安全性和软件的安全性关系，如果“可执行文件”的格式设计得更加科学合理，那么将更难感染病毒，从而提高软件的安全性；“可执行文件”的类型和格式将影响软件开发过程以及相应的成本，比如脚本开发的效率可能比目标程序文件的开发效率要高，但是从另外一个层面来说，前者的运行效率就一般比目标程序文件的运行效率低，进而影响软件的工作效率；除此之外，“可执行文件”的大小将直接影响软件的应用领域，比如嵌入式领域希望使用占用更少字节的“可执行文件”；除此之外，“可执行文件”的格式可能还影响到软件的部署和软件设计的架构。

因此，可以这么说，软件领域所面临的问题最终都将归结到“可执行文件”上去，因为它是软件的最终载体。

但是“可执行文件”的各种属性与软件领域所面临的问题之间到底存在怎样的关联呢？这可能是解决问题的突破口，而这正是计算机科学与技术的一个分支——数据挖掘技术关注的内容。

下面我们将分析引入数据挖掘技术来寻求相关问题解决途径的可行性。

4. 引入数据挖掘技术的可行性分析

“数据挖掘是从数据库中识别出有效的、新颖的、潜在有用的并且最终可理解模式的非平凡过程。”[3]

数据挖掘作为知识发现过程的一个基本步骤[4]，它涉及多种学科和技术，包括数据库和数据仓库技术、统计学、机器学习、高性能计算、模式识别、神经网络、数据可视化、信息检索、图像与信号处理以及空间或时间数据分析。它可以对多种数据类型，包括关系数据库、数据仓库、事务数据库、时间数据库、空间数据库等采用智能方式提取模式，这些模式有“概念 / 类描述：特征化和区分”，“频繁模式、关联和相关”，“分类和预测”，“聚类分析”，“离群点分析”，“演变分析”等。

而在计算机科学与技术日益蓬勃的今天，已经积累了大量的与“可执行文件”相关的数据和技术，这些数据包括各种不同操作系统平台上运行的具有各种不同格式和功能，并且采用了不同算法设计思想，拥有不同的运行效率的软件副本和相关文档。因此可以根据这些数据的类型以关系数据库、数据仓库、时间数据库、空间数据库等各种不同方式存储起来，并结合各种成熟的软件技术辅助数据挖掘的过程，进而提取到有用的模式，为解决相关的问题提供依据和途径。

下面是我们列举几个可行的数据模式。

4.1 特征化和区分

病毒是一种典型的“可执行文件”，它具有破坏性，对软件安全构成威胁，如何来降低这种威胁呢？

特征码查杀技术是用来监测病毒的最简单、开销最小的方法，原理是将所有的病毒加以剖析，将病毒独有的特征收集在一个病毒资料库，也就是病毒库，以扫描的方式将监测程序与病毒库的特征码进行一一对比，如果发现相同的代码，就判定待测程序已感染病毒。

但是特征码技术明显存在一些弊端[5]，比如滞后于病毒的出现，在电脑病毒数量越来

越多、传播速度越来越快的网络环境下，将难以满足用户的安全需求。

从特征码技术的工作原理可以看出它是数据挖掘中的“特征码和区分”模式的典型实例。如果能够结合数据挖掘技术的相关成果，比如利用数据仓库的异构数据源的特性和一些相关的算法来提高病毒特征码的提取和病毒库的更新速度，并结合其他病毒查杀技术，比如虚拟机技术，就有可能更加有效的防毒杀毒。

4.2 频繁模式、关联和相关

“可执行文件”的诸多属性之间可能存在很大的关联，虽然目前有大量的“可执行文件”以及它们的属性信息，但是它们并没有被整合和利用起来，因此并不能够找到足够有用的信息帮助我们解决相关的问题。如果能够结合数据挖掘技术，把当前“可执行文件”的大量数据信息进行数据清理、集成、选择、变换操作，进而结合一些成熟的数据挖掘算法提取出有效的数据模式，并进行进一步的评估和知识表示，可能能够对相关问题起到极大的帮助。

这其中一个可能的应用是寻找“可执行文件”的各个属性之间的关系，发现同时出现几率最大的相关属性这一数据挖掘的“频繁模式、关联和相关”模式。

例如，如果能够找出代码运行效率和可执行文件大小之间的关系，那么可能找出提高代码运行效率和减少文件大小的途径。比如，代码的运行效率可能反应于代码的覆盖率，而获取到代码的覆盖率就可以找出那些没有被执行的代码，从而找出优化代码、剔除未执行代码的途径，进而减少“可执行文件”的大小，这样就可以提高软件在嵌入式领域的应用。又如，如果能够找出各种可执行文件格式和它们感染病毒几率之间的关联就可能找出具有更高安全属性的可执行文件格式以及它们的特点，进而辅助设计出更安全的可执行文件格式。

这样的话，可以考虑结合数据挖掘技术和现有的软件技术，比如软件开发、设计、测试、维护、安全等技术等进行“频繁模式、关联和相关”方面的挖掘，找出提高软件质量和拓展软件应用领域的潜在途径。

4.3 演变分析和预测

又回到软件安全问题，在病毒查杀领域，一个比较重要的问题是病毒可能会不断的变异，不断出现新的变种，它们的特征码可能一直在发生变化，因此，如果仅仅采用传统的特征码查杀技术可能会受到很多限制和处于被动状态。如果能够利用数据挖掘技术寻找病毒变异的规律和趋势，那么就可能能够预测病毒的变种，做出提前的预报和提出可能的防范措施。

通过上述分析，我们发现，引入数据挖掘技术来寻求软件领域所面临问题的解决途径是可行的。

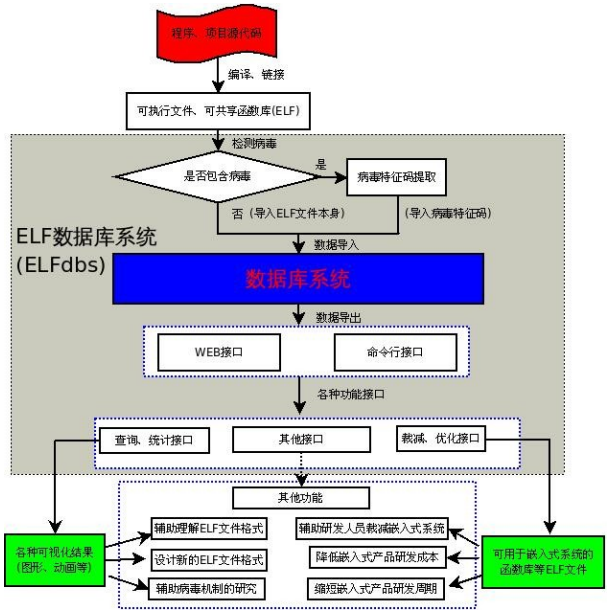
下面来介绍一个简单的实例。

5. 实例简介

ELF(Executable and Linking Format)[8] 是 UNIX 平台上的一种通用可执行文件格式，它的数据结构跟关系数据库很相似，因此，容易被存储到关系数据库中，这样就可以结合数挖掘技术进一步进行模式提取、评估和知识表示等工作。

ELFdbs 项目基于此而计划打造一个 ELF 文件的数据挖掘平台，为用户提供 ELF 文件导入并提供相关查询、统计、分析、裁减性导出等接口。

这里是 ELFdbs 的大概工作流程：



该项目目前处于前期调研和准备阶段，预计实现的功能有：

- 辅助程序员理解可执行文件格式，进而理解程序开发的生命周期，比如以图形化的方式展示可执行文件的内部结构。
- 打造一个病毒技术研究的平台，比如进行病毒特征码的提取和分析，建立一个病毒数据仓库，为病毒查杀技术提供远程支持等。
- 辅助下一代可执行文件格式的研究，比如提供有效的 ELF 可执行文件的查询、统计、分析平台，产生有效的数据报告。
- 辅助嵌入式开发人员裁减和获取特定的函数库，比如提供一个接口，只需要用户输入特定的函数列表，就可以导出一个特定的函数库，进而极大的提高嵌入式开发人员的工作效率，降低嵌入式系统开发的成本。

6. 总结

通过本文的初步探索，我们发现，结合现有的各种软件技术，对大量的与“可执行文件”相关的数据进行一定的数据挖掘处理和分析，可能寻找到计算机软件领域所面临的问题的解决途径，进而达到提高软件的开发效率和运行效率，拓展软件的应用领域，提高软件的安全性和可移植性等目标。

参考资料

[1] http://en.wikipedia.org/wiki/Executable

[2]https://engineering.purdue.edu/ECN/Support/KB/Docs/GlossaryE

[3] 生物信息数据挖掘技术的典型应用
[4] 数据挖掘概念与技术
[5] 特征码查杀技术
http://baike.baidu.com/view/595316.htm
[6] 泛谈虚拟机及其在反病毒技术中的应用
http://it.rising.com.cn/antivirus/rviruslore/rvirus028.htm
[7] 什么叫代码覆盖率
http://blog.csdn.net/Kesa_Kong/archive/2007/06/14/1652341.aspx
[8] Executable and Linking Format
http://www.skyfree.org/linux/references/ELF_Format.pdf