第11章 软件老化可信度量模型

陈仪香

华东师范大学软件学院可信智能团队TrIG

2024年12月18日



第11章 软件老化可信度量模型

11.1 软件老化

11.2 老化类别

11.3 度量元模型

11.4 可信计算框架



软件老化是一种在经历长时间运行后的系统中极有可能出现的现象, 主要表现为软件系统在长时间运行后,受内存泄漏、碎片问题、数 值累计错误等原因影响,会产生软件状态异常或性能下降的情况, 情况严重时甚至会出现软件失效或系统宕机[1]。

该现象是在许多软件系统中根据经验观察得出的,一般特征是随着系统或进程的运行时间增加,其故障率也会增加[2],而引起软件老化的缺陷统称为 Aging-Related Bug (ARB)。

[1] 孟令泽. 基于人工蜂群算法与 BP 神经网络模型的云软件老化预测机制研究[D]. 内蒙古大学,2020.

[2] Michael Grottke, Rivalino Matias, and Kishor S Trivedi. The fundamentals of software aging. In 2008 IEEE International conference on software reliability engineering workshops (ISSRE Wksp), pages 1–6. Ieee, 2008.

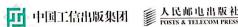
- ●失信证据是指可以反映软件运行结果偏 离预期或性能下降的相关指标。
- ●由于与运行时间相关的 bug 在所有 bug 中占比非常小,所以需要从大量错 误报告中提取失信证据。
- ●为了使获取的失信证据涉及面更广、反 映问题更全面,将《致命 bug》中所 述历史上知名的软件事故。







软件缺陷的灾难与启示





《致命Bug:软件缺陷的灾难与启示》 [韩] 金钟河 著,叶蕾蕾 译

编号	缺陷 名称	缺陷描述	国别年份
1	爱国 者导 弹	0.000000095的误差夺走28条生命:系统舍去二进制的25位以下的数字,只保存24位,所以每运行1个clock,就会产生(按照十进制的话约为0.00000095)的误差。	美国 1991
2	火星 探测 器	火星气候轨道探测器MCO在着陆过程中,MCO向地面站传回发动机推力信息时,使用国际标准单位"牛顿",而地面站使用的程序是SM_FORCE,使用的却是美国通用的力的单位磅力,1磅力约等于4.45牛顿。	美国 1999
3	阿丽亚娜5	火箭发射37秒后,发生爆炸。测定当前火箭高度和速度组件SRI,阿5使用阿4的SRI。规定输入SRI的数值是16位元整数。但当时输入SRI的数值是64位元浮点数,因而SRI是无法使用16位元整数表示的。用SRI的16位元整数形式对该数值进行转换过程造成了溢出错误。无法发出正确的火箭高度和速度。	欧洲 1996



- Grottke 等人 研究使用的四个带有公共漏洞库和活跃漏洞库的开源软件系统作为研究对象,在前人研究的基础上进一步探究 bug 与软件老化之间的联系,提取失信证据。这四个拥有众多用户的软件系统涵盖了不同类型的软件,可以从中获取大量错误报告来提取失信证据:
- 1. Linux 内核,用于从嵌入式系统到超级计算机等多个领域,是一个功能丰富的操作系统; https://bugzilla.kernel.org (Linux 2.6),
- 2. MySQL,占有重要的市场份额,是最常用的数据库管理系统之一; http://bugs.mysql.com (MySQL 5.1)
- 3. Apache HTTPD 服务器,为用户提供动态 Web 服务,是 Apache 产品线下著 名的一款 web 服务器; https://issues.apache.org/bugzilla (Apache HTTPD 2),
- 4. Apache AXIS 框架,许多公司采用其 Web 服务来运行 Web 应用程序,是一个开源、基于 XML 的 Web 服务架构。

https://issues.apache.org/jira/secure/IssueNavigator.jspax (Axis 1)



- Grottke 等人 研究使用的四个带有公共漏洞库和活跃漏洞库的开源软件系统作为研究对象, 在前人研究的基础上进一步探究 bug 与软件老化之间的联系,提取失信证据。这四个拥有 众多用户的软件系统涵盖了不同类型的软件,可以从中获取大量错误报告来提取失信证据:
- 1. Linux 内核,用于从嵌入式系统到超级计算机等多个领域,是一个功能丰富的操作系统;(Linux 2.6)
- 2. MySQL,占有重要的市场份额,是最常用的数据库管理系统之一; (MySQL 5.1)
- 3. Apache HTTPD 服务器,为用户提供动态 Web 服务,是 Apache 产品线 下著 名的一款 web 服务器;(Apache HTTPD 2)
- 4. Apache AXIS 框架,许多公司采用其 Web 服务来运行 Web 应用程序,是 一 个开源、基于 XML 的 Web 服务架构。(Apache Axis 1)



●例 3.1 根据 Linux 缺陷库中 Bug3171 的记载,在 2004 年 Linux 的日志文件系统 ext3 运行的一段时间后自己的磁盘空间所剩无几,这是一个与运行时间相关的 bug。详细研究错误报告发现,内核本应在 ext3 删除日志后回收相应的磁盘空间,但是由于源码中存在单行嵌套错误,文件句柄一直没有关闭导致回收失败,进而造成这次磁盘空间泄露,得到表X.1。

出处	描述	失信原因	失信证据
LinuxBug3171	Linux 的日志文	磁盘空间泄漏导	软件占用预期之外的
	件系统 ext3 运	致磁盘空间占用、	磁盘空间占总空间的比
	行的一段时间后	虚拟内存不足、	对系统性能产生影响
	发生磁盘泄漏	系统效率降低	所需的时间



●例 3.2 根据《致命 bug》中温哥华证券交易所事件的记载,在 1982 年初温哥华 证券交易所推出了初始设定为 1000 点的股票指数。该所每天要对股票进行 2800 次计算和更新,由于程序编写失误导致计算时会直接舍去小数点后三位而非四 舍五入,即使一次计算只产生 0.001 的误差,积少成多后导致股票指数每天都会下滑一到两个百分点,一年后股票指数几乎缩水一半。造成这次重大软件事故的 正是舍入误差的累积,由此得到表X.2。

出处	描述	失信原因	失信证据
温哥华证券交易所事件	温哥华交易所推出的股票指数一年后缩水近一半	微小的运算操作误差 经过较长时间的累积 导致结果偏离预期	



●从 4 大缺陷库中统计的 bug 数总计 963 个,经过筛选,共得 55 个与软件老化有关的错误报告,结合《致命 bug》 中涉及的 18 个著名的软件事故,其中有 4 个与软件老化有关,以此作为设计度量元的依据。

第11章 软件老化可信度量模型

11.1 软件老化

11.2 老化类别

11.3 度量元模型

11.4 可信计算框架



老化类别

基于软件老化的失信证据度量元除了表现形式外,也可以从导致软件老化的原因角度进行研究。参考 Cotroneo 等人 对软件老化相关 bug 的分类方法,将度量元按失信的成因进行分类,分为 5 类,如表X.9,

软件老化原因分类	描述
MEM (内存管理)	度量元导致与内存管理有关的错误积累(例内存泄漏,未刷新缓冲区)
STO(存储空间)	度量元导致影响存储空间的错误累积(例如,错误占用磁盘空间)
LOG (逻辑资源)	度量元导致"其他逻辑资源"泄露,即依赖于系统的数据结构 (例如,
	使用未释放的套接字或索引节点)
NUM (误差累积)	度量元导致数字误差累积(例如,舍入误差,整数溢出)
ARU(未知原因	度量元仅知道发生错误的该率会随着时间而增加(例如,因为它仅在 多次调用某个功能之后才导致失败),或是没有足够的关于确切失败
ARB's Subtype: Unknown	机制的信息,无法找到明确的原因



老化类别

ID	A-TYPE	REASON
linux3171	STO	单行 {} 嵌套错误导致大量磁盘空间泄漏
linux5137	MEM	驱动程序错误,巨大的 fifo 溢出
linux7959	ARU	Tg3 驱动程序存在挂起/恢复问题
mysql34335	NUM	主键自动增长时 bigint 溢出崩溃
mysql34335 NUM 主		主键自动增长时 bigint 溢出崩溃
		截断后不刷新查询缓存,如果表不是空的,
mysql40386	LOG	record 的值永远不能是 0



老化类别

ID	A-TYPE	REASON		
mysql46656	MEM	删除表时会发生内存泄漏,关机时没有释放内存		
		可用内存检查使崩溃恢复速度降低数十倍,		
mysql49535	MEM	在读取崩溃恢复日志的过程中使用超过 90%的 CPU		
mysq156340	LOG	非索引更新后, innodb 过于频繁地更新索引统计信息		
		如果日志文件达到 2GB, Apache 会失灵。 错误日志		
apache13511	STO	达到 2G 后就无法写入了		
apache29962	MEM	字节范围过滤器将缓冲整个响应在了内存中		
		SOAPHeaderElement.detachNode()中的空指针异常。		
AXIS-1248	ARU	第二次调用 detachNode()后,将引发空指针异常。		

第11章 软件老化可信度量模型

11.1 软件老化

11.2 老化类别

11.3 度量元模型

11.4 可信计算框架



失信证据是指可以反映软件运行结果偏离预期或性能下降的相关指标。

失信证据对应的风险值越大, 风险等级也就越高,可能导致软件进入预期之外状态的种类也就越多,表X.4给出风险等级以及可能导致的最坏风险 状态的映射关系。

风险等级	风险值	风险状态
V	10	运行时长会显著影响软件运行结果
IV	9	运行时长有大概率影响软件运行结果或对性能产生影响
III	7	运行时长有概率影响软件运行结果或对性能产生影响
II	4	运行时长对运行结果没有影响,但是有概率对软件性能产生影响
Ι	1	运行时长对软件没有影响



华东师范大学软件学院可信智能团队

Trustworthy Intelligence Group

鉴于风险等级越高,可能导致的异常状态数就越多,平均到单个风险状态对 风险值的影响应逐渐降低,因此将风险值按近似黄金分割的比例逐层递减直至 失信证据对软件运行没有影响,将此时的风险值设为 1。还是以 LinuxBug3171 为例,如表X.5。

出处	度量元名称	失信原因	失信证据	风险等级对应	立指标	风险值
			软件占用预	87%-100%	V级	10
		磁盘空间	期之外的磁盘	59%-87%	IV级	9
		泄漏导致 温盘 出	空间占总空间	400/ 500/	III 级	7
	磁盘空间泄露		磁盘空间 的比	20%-40%	II 级	4
Linux3				0-20%	I级	1
171				0-10	V级	10
			- 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	10-30	IV级	9
				20 (0	III 级	7
				60-110	II 级	4
			Մա Մուլնուրը այ	>110	I级	1



华东师范大学软件学院可信智能团<mark>队</mark>

Trustworthy Intelligence Group

当 Linux 的日志文件系统运行 48 小时后, 软件就会额外占用系统 80% 的磁 盘空间, 导致系统运行失常, 这里两个失信证据分别为 80% 和 48 小时, 对应了 IV 级和 III 级的风险等级, 两个失信证据对应的风险值就为 9 和 7。

出处	度量元名称	失信原因	失信证据	风险等级对应	立指标	风险值	
			软件占用预	87%-100%	V级	10	
		磁盘空间	期之外的磁盘	59%-87%	IV 级	9	
		泄漏导致 温盘 出	泄漏导致磁盘交回	空间占总空间	400/ 500/	III 级	7
	磁盘空间泄露				20%-40%	II 级	4
Linux3			はろたび	0-20%	I级	1	
171				0-10	V级	10	
			/ 1 /1 /7 / D 1 - 1 1 1	10-30	IV 级	9	
				20 (0	III 级	7	
				60-110	II 级	4	
			MT LAI CACD CHT -	>110	I级	1	
<u> </u>							

基于软件老化的失信证据度量元是指可能导致软件老化的可被度量的 直接或间接软件项目元素,通常在软件开发和运维服务过程中获取,用六元组表示:

Metric element based on evidence=<Name,M-Type,Reason,A-Type,Evidence,R-Value>

其中,

- ◆ Name 表示度量元的名称;
- ◆ M-Type 表示度量元的类型;
- ◆ Reason 表示度量元 失信的原因;
- ◆ A-Type 表示软件老化原因的类型;
- ◆ Evidence 表示失信证据;
- ◆ R-Value 表示失信证据的风险值。

软件老化的表现形式多种多样,有些表现为运行中由于涉及某些操作导致系统崩溃,有些则是在运行一段时间后突然宕机,还有的可能是随着运行时间的 增加系统性能不断下降等等。

为了覆盖所有可能的情况, 设计了三种类型的 度量元, 分别为:

触发型度量元:指只有在特定情况下才会触发问题的软件项目元素。

增长型度量元:指随运行时间增长,原本影响较小却可不断累积的软件项目元素。

突变型度量元:是指某些指标随时间增长至临界点时会发生突变的软件项目元素。



触发型度量元:指只有在特定情况下才会触发问题的软件项目元素。

例如 "LinuxBug11935 权限管理错误",软件运行中遇到需要对文件进行读写的情况, 而该文件状态为只读,导致软件无法正常运行。具体如表X.6。

表 X.6 触发型度量元分类示例

度量元名称 度	度量元类型	失信原因	失信证据
权限管理错 触误		系统无法访问自己本该 被授权的资源或访问自 己未被授权的资源	操作相关文件状态不满足规定要求的比例



增长型度量元:指随运行时间增长,原本影响较小却可不断累积的软件项目元素。例如"MySQLBug46656 删除表未释放内存",在使用MySQL的 delete 操作来删除数据文件后,占用的空间没有被完全释放,经过长时间的运行被占用的空间就会积累到不可忽视的地步。

经过研究发现, 其实 delete 操作只是指挥系 统只是删除了数据文件对应的标识位, 而具体文件依旧存在于数据库中, 导致空 间仍然被占用, 只有新写入数据时才会进行覆盖, 且可能留下数据碎片, 无法完 全释放空间, 具体如表X.7。

度量元名称	度量元类型	失信原因	失信证据
删除表未释放内存	增长型	delete 操作只是指挥系统只是删除了数据文件对应的标识位,而具体文件依旧存在于数据库中,导致空间仍然被占用,只有新写入数据时才会进行覆盖,且可能留下数据碎片,无法完全释放空间。	程序固定占用内 存之比



突变型度量元: 是指某些指标随时间增长至临界点时会发生突变的软件项目 元素。

例如"MySQLBug34335 数值溢出",数据库中自动增长的编号在运行一段时间后大于数据类型上限,软件失效,具体如表X.8。

度量元名称	度量元类型	失信原因	失信证据
数值溢出	突变型	当自动增长的数据 大于当前数据类型 上限导致数据失效	込到临乔诅干均所需的



通过对提取的失信证据按两种分类方法同时进行研究,去除重复、 冗余的 部分,设计出最终的度量元模型,与定义的六元组对应,部 分度量元展示如下 表X.11,完整的度量元表格。

ID	度量元	度量元	失信原因	软件老化	失信证据	风险等级对应	风险	出处
	名称	类型		原因分类		指标(V-I 递减)	值	
							r(t)	
			24位的固		48 小时	结果完全不可信	10	
1	浮点数	增长	定小数点 寄存器会	ARU	后误差对 预期结果	影响较大	9	爱国
	寄 存	型	可任命云 舍去 25位		的影响	有影响结果个リ	7	者导
			以下的数			靠		, , ,
	器误差		字,导致			影响结果小,可	4	弹事
			难以避免			接受		件
			的误差			无影响	1	,



ID	度量元	度量元类	失信原因	软件老化原	失信证据	风险等级对应指标	风险值	出处
	名称	型		因分类		(V-I 递减)	r(t)	
		触	程序所需内存		程序运行 需	88%-100%	10	
	内存	发	大于系统可用 内存可能导致		 要占用 的内	59%-88%	9	
2	需 求 过大	型	预期功能无法	10112101	存空 间占总	40%-59%	7	Linux2425,
			实现或系统效 率降低		内 存的比	20%-40%	4	Linux7536
			+r4 IX			0-20%	1	
					程序所需 索	88%-100%	10	
						59%-88%	9	
	索引节				引节点 占系	40%-59%	7	
3		突	索引节点用尽		统拥 有索引	20%-40%	4	
	点爆满	变 型	时系统无法再		节 点的比例	0-20%	1	
		(全)	创建新目录或		达到临界	0-10	10	Linux3431
			文件		值平均所	10-30	9	Lillux3431
					需的时间	30-60	7	
						60-110	4	
					(h)	>110	1	



ID	度量元	度量元	失信原因	软件老化	失信证据	风险等级对应指	风险值	出处
	名称	类型		原因分类		标(V-I 递减)	r(t)	
			系统无法访问		操作相关	87%-100%	10	
		角虫	自己本该被授		文件状态	59%-87%	9 7	
4	权限管	发		MEM	不满足规	0-20%	4	Linux11935
'	理错误		权的资源或访		定要求的	0-2076	1	Linuxiiiy
		型	问自己未被授		比例			
			权的资源					
			缓冲区本身的容		达到临界	0-10	10	
		突	量有限,在写入		值平均所	10-30	9	
	缓冲区				需的时间	30-60	7	Linux5137,
5		变	数 据时可能一	MEM	(h)	60-110	4	Linux5711,
	溢出	型	次超出 了上限,		(11)	>110	1	AXIS1270
			造成溢出 甚至					111101210
			覆盖其他数据					



ID	度量	度量元	失信原因	软件老	失信证据	风险等级对应	风险值	出处
	元	类型		化 原因		指标(V-I 递	r(t)	
	名称			分类		减)		
		突	当自动增长		达到临界	0-10	10	
6		变	的 数据大于		值平均所	10-30	9	MySQL
	数值	型	当前 数据类	N TT TN /	需的时间	30-60	7	
			型上限 导致	NUM	(h)	60-110	4	34335
	溢出		数据失效			>110	1	
		触	多个进程因			87%-100%	10	MySQL
7	死锁	发	竞争共享资	LOG	触发死锁	59%-87%	9	52814,
		型	源而处于永		的概率	20%-59%	7	
			远等待的状态			0-20%	4	AXIS136
			态			0	1	9



一个度量元可以由多个失信证据支撑,对应多个风险值,该度量元最终风险值取其中的最大值,因为只要有一个失信证据风险值偏大就说明该度量元导致软件进入预期之外运行状态数较多,发生软件老化的可能性就大。

表格中指标的 设计源于超过两百位编程人员的问卷调查,在对问卷结果进行整理汇总的基础 上结合实际情况,最终确定为上述数值,具体的问卷情况可查阅附录。

文中最终 设计了共计 30 个基于软件老化的失信证据度量元,后续可扩充取材范围进行添 加,本章仅提供设计思路以供参考。



东师范大学软件学院可信智能团队

Trustworthy Intelligence Group

ID	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
			程序所需内存大于		程序运行需要占用	88%-100% 59%-88%	10 9	Linux2425, Linux7536 Linux5144, Linux9468, Linux13293, MySQL49535, MySQL48993, Apache12320, Apache27751 Linux11935 Linux34622
M1	上去房子 斗上	&고 42- 표미	系统可用内存可能	MEM	的内存空间占总	40%-59%	7	
	内存需求过大	触发型	导致预期功能无法	MEM		20%-40%	4	
			实现或系统效率降低		内存的比	0-20%	1	Linux1330
						88%-100%	10	
					长时间占用的内存大小	59%-88%	9	Linux5144,
					占总内存的比	40%-59%	7	Linux5144, Linux9468, Linux13293, MySQL49535, MySQL48993, Apache12320, Apache27751
			高优先级或其他资源	MEM	白心的行的几	20%-40%	4	Linux13293,
	资源占用	触发型	长时间占用资源			0-20%	1	Linux7536 Linux5144, Linux9468, Linux13293, MySQL49535, MySQL48993, Apache12320, Apache27751 Linux11935
M2	火炒口/1	瓜久王	导致资源无法释放			00-10	10	1 * '
			寻致贞源儿宏样成		对系统性能产生影响	10-30	9	1 1
					平均所需的时间	30-60	7	
					1 **3//1110 #3#31/4	60-110	4	Apacheziroi
						>110	1	
			系统无法访问自己本			87%-100%	10	Apache27751 1 10 9 7 4 Linux11935
			<u> </u>		操作相关文件状态	59%-87%	9	
110	权限管理错误	触发型	该被授权的资源或访	MEM	JULI INVESCIT VICE	20%-59%	7	
M3					不满足规定要求的比例	0-20%	4	
			问自己未被授权的资源			0	1	Linux5144, Linux9468, Linux13293, MySQL49535, MySQL48993, Apache12320, Apache27751 Linux11935
						88%-100%	10	
					规定时间内程序	58%-88%	9	
						40%-58%	7	Linux9468, Linux13293, MySQL49535, MySQL48993, Apache12320, Apache27751 Linux11935
					额外占用内存大小	19%-40%	4	
		124 17 174	程序自我复制超出	100		0-19%	1	
M4	程序异常繁殖	增长型	预期,占用资源过多	MEM		00-10	10	Linux34622
			377917 H/H X VAX 29		对系统性能产生影响	10-30	9	
						30-60	7	
					平均所需的时间	60-110	4	
						>110	1	



ID	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
			当一个对象不存在时又调			>11	10	
			用其方法会产生异常,			6-11	9	Linux1209,
M5	空指针相关问题	触发型		MEM	指针调用不当数目	3-6	7	Linux6043,
			当访问或修改一个对象不			1-3	4	AXIS1248
			存在的字段时会产生异常			0	1	NX101240
			当内存输入超出了预分配			30%-100%	10	Linux6114,
			的空间大小,就会覆盖该		出现大于预分配空间	15%-30%	9	Linux0114,
M6	内存越界	触发型		MEM	山 <u></u> 近 八 八 八 1 八 八 1 1 1 1 1 1 1 1 1 1 1 1 1	5%-15%	7	MySQL38191,
			空间之后的一段存储区域,		大小的输入的概率	0-5%	4	"火星全球勘测
		导3	导致系统异常			0	1	者"号事件

			向缓冲区填充数据超其			00-10	10	
			本身的容量,而导致数据		达到临界值平均	10-30	9	Linux5137,
M7	缓冲区溢出	突变型	溢出到被分配空间之外的	MEM		30-60	7	Linux5711,
			内存空间,溢出的数据覆		所需的时间(h)	60-110	4	AXIS1270
			盖其他内存数据			>110	1	



ID	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
			delete 操作只是指挥			88%-100%	10	
			系统只是删除了数据		实际使用内存与程序 固定占用内	58%-88%	9	
					存之比	40%-58%	7	
			文件对应的标识位,		行之比	19%-40%	4	
M	删除表未释放内存	增长型	而具体文件依旧存在	MEM		0-19%	1	MySQL466
8			于数据库中,导致空			0-10	10	56
			间仍然被占用,只有		对系统性能产生影响 平均所需的			
			新写入数据时才会进		时间	10-30	9	
			行覆盖,且可能留下			30-60	7	
						60-110	4	
			数据碎片,无法完全			>110	1	
			释放空间。					
						55%-100%	10	
	31四十年份	操化型	使用完的引用由于引用		コロナ油畑 トル			
M 9	引用未清空	增长型	未及时清空导致无法	MEM	引用未清理占比			AXIS12 02
9			自动回收内存					
						30%-55%	9	
						15%-30%	7	_
						5%-15%	4	_
	<u> </u>	<u> </u>		<u> </u>		0-5%	1	
						>100 50-100	10	Linux5284,
M	a 循环过度	突变型	循环过度导致运行时间	MEM	循环复杂度	30-50	7	2003 年美
10	1/月217年/文	八文王	过程或者内存被全部占用	IVILLIVI	/////////////////////////////////////	10-30	4	— 国东北部
10						1-10	1	大停电
								事件



ID	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
						>200m	10	
						150m-200m	9	MySQL45989 , MySQL33247 MySQL32709
			只要你去与我们提供土地		未释放内存大小	100m-150m	7	
М	异常终止	增长型	异常终止后部分操作未执 行,导致程序程序崩溃	MEM		50m-100m	4	MCOI 45000
11			11) 4 20/11/11/11/11/07			0-50m	1	MySQL43989
						0-10	10	MySQL33247
						10-30	9	
					对系统运行产生影响 平均所需的时	30-60	7	
					间	60-110	4	
							4	
						>110		
						>200m	10	
					+ *** *** + **	150m-200m	9	
					未释放内存大小	100m-150m	7	
М			回滚机制设计不完 善,			50m-100m	4	
12	回滚机制问题	增长型	导致资源浪费	MEM		0-50m	1	MySQL32709
						0-10	10	1
						10-30	9	
					对系统运行产生影响 平均所需的时	30-60	7	
					间	60-110	, 4	
						>110	1	
						/110		



II	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
						>200m	10	
						150m-200m	9	Apache9168 Linux11100, Linux11377, Apache9708, Apache29962 MySQL56709 Apache14098, Apache24991, Apache26562, Apache27106, AXIS1423, AXIS2278
					占用的内存大小	100m-150m	7	
	原口出和计名	操化型	大量僵尸进程占用 系统			50m-100m	4	T
13		增长型	资源,影响性能	MEM		0-50m	1	Apache9168
13						0-10	10	
					 对系统运行产生影响 平均所需的	10-30	9	
					时间	30-60	7	
						60-110	4	
						>110	1	
						87%-100%	10	Linux11100,
			缓存一致性、缓存并发、 ^{这型} 缓存穿透、缓存的雪崩现 象	МЕМ	一次性操作数据的规模 EM 占可用资源的比	59%-87%	9	Linuv11277
M 14	高并发缓存问题	触发型				40%-59%	7	
17			象			20%-40%	4	Apache9708,
						0-20%	1	Apache29962
M 15		增长型	用动态存储分配函数动态 开辟的空间,在使用完毕 后未释放,结果导致一直 占据该内存单元,直到程 序结束	MEM	平均每次使用软件增加 的额外内存占总内存 的百分比	55%-100% 30%-55% 15%-30% 5%-15% 0-5%	10 9 7 4 1	Apache14098, Apache24991, Apache26562, Apache27106,

度量元设计(STO4)



ID	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
					软件占用预期之外的磁	87%-100% 59%-87%	10	
						40%-59%	10 9 7 4 1 10 9 7 4 11 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 9 7 4 1 10 MySQL52964	
			磁盘空间泄漏导致		盘空间占总空间的比	20%-40%	4	
	~// 선 수 / July 프	126 14 754	磁盘空间占用、			0-20%	1	Linux3171
S1	磁盘空间泄露	增长型	虚拟内存不足、	ST0		00-10	10	Linux3171
					对系统性能产生影	10-30		
			系统效率降低			30-60	7	
					响平均所需的时间	60-110	4	
						>110	1	
					40 -> 40 -> -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1	88%-100%	10	
					程序所需索引节点数	59%-88%	9	
	2 索引节点爆满 3			占系统排	占系统拥有索引节点	40%-59%	7	
			索引节点用尽时		数的比例	20%-40%	4	1
CO	索引节点爆满	र्द्ध केंद्र संग		ST0	30,100,1	0-20%	1	1 . 0401
52	系列卫思漆俩	突变型		510		00-10	10	Linux3431
			新目录或文件		达到临界值平均	10-30	9	
						30-60	7	
					所需的时间(h)	60-110	4	
						>110	1	
S3	数据插入问题	触发型	数据插入方式错误 或插入 位置错误,导致缓冲区或 者磁盘空间被填满		触发插入问题的概率	88%-100%	10	MySOL52964
				510		58%-88%	9	
						19%-58%	7	
						0-19%	4	
						0	1	

度量元设计(STO4)



Ι	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值r(t)	出处
S	4 写入失败	突变型	日志文件过多未及时清理, 导致空间被占满, 新日志无法写入	STO	达到临界值平均 所需的时间(h)	0-10	10	Apache135
						10-30	9	
						30-60 60-110	4	
						>110	1	

度量元设计(LOG7)



ID	度量元名称	度量元类型	失信原因	软件老化原因分 类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
L1	嵌套字功能异常	触发型	发生套接字超时或	LOG	嵌套字功能异常的概率	86%-100%	10	Linux32832, AXIS769
			其他嵌套字功能			59%-86%	9	
						20%-59%	7	
						0-20%	4	
Щ			异常时线程阻塞			0	1	
L2	未刷新缓存	触发型	缓存未及时刷新导致 结 果偏离预期	LOG	截断后缓存未刷新	1	10	MySQL403 86
						0	1	
L3	死锁	触发型	多个进程因竞争共 享资源而处于永远 等待的状态	LOG	触发死锁的概率	87%-100%	10	MySQL52814, AXIS1369
						59%-87%	9	
						20%-59%	7	
						0-20%	4	
						0	1	
L4	无效更新	触发型	重复自动更新变化 内容, 或更新索 引时未发生变 化 导致资源浪费	LOG	无效更新的占比	87%-100%	10	MySQL56340
						59%-87%	9	
						40%-59%	7	
						20%-40%	4	
						0-20%	1	
	。 临时文件未及时清理	增长型	临时文件未及时清理, 导 致占用过多内存	LOG	占用的内存大小	>200m	10	MySQL40013
						150m-200m	9	
T 5						100m-150m	7	
LO						50m-100m	4	
						0m-50m	1	

度量元设计(LOG7)



华东师范大学软件学院可信智能团队 ▲ Trustworthy Intelligence Group

ID	度量元名称	度量元类型	失信原因	软件老化原因分类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
L6	适配器选择错误	触发型	适配器模式中接口 选择	LOG	适配器选择不当	1	10	Linux7718
			不当			0	1	
						0-10	10	
L	嵌套字用尽	突变型	嵌套字资源用尽导 致无	LOG	达到临界值平均所需的时间(h)	10-30	9	AXIS238
7	MA 1711	八人工	嵌套字可用	LOG		30-60	7	717115250
′						60-110	4	
						>110	1	

度量元设计(NUM2)



_	Trustworthy	Intel	ligen	ce Grou	ın
	ソンシャンロン く ユーゼ		170		

II	度量元名称	度量元类型	失信原因	软件老化原因分类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
N 1	数值溢出	突变型	当自动增长的数据 大于当前数据类型上限 导致数据失效	NUM	达到临界值平均 所需的时间(h)	0-10	10	MySQL34335
						10-30	9	
						30-60	7	-
						60-110	4	-
						>110	1	-
			 运算中舍入操作经过			一个月以内	10	Linux20882,
N	2 舍入误差累计	增长型	较长时间的累积导致	NUM	误差对结果产生不可 忽视的影响所需的时间	一个月至一年	9	温哥华证券 交易所事件
			误差巨大			一至十年 十年以内	4	
L						一直无影响	1	

度量元设计(ARU2)



东师范大学软件学院可信智能团队 Trustworthy Intelligence Group

ID	度量元名称	度量元类型	失信原因	软件老化原因分类	失信证据	风险等级对应指标(V-I递减)	风险值 <mark>r(t)</mark>	出处
A	浮点数寄存器误差	增长型	24 位的固定小数点寄存 器 会舍去 25 位以下的数 字, 导致难以避免的误差		48 小时后误差对 预期结果的 影响	结果完全不可信	10	爱国者导 弹事件
			1.以作以起光的《左			影响较大 有影响结果不可靠	9	- 31 4.11
						影响小结果可接受	4	-
						无影响 0-10	10	
			仅指系统连续运行一段时间		出现异常平均 所需的时间			Linux3134, - Linux5870,
A2	系统功能异常	触发型	后,系统性能下降或 功能出现异常,但具体原因	ARU		10-30 30-60	9 7	Linux7959,
			无从考证的情况			60-110 >110	1	
					发生功能异常 的概率	30%-100%	10	
						15%-30%	9	<u> </u>
						5%-15% 0-5%	4	-
						0	1	

第11章 软件老化可信度量

11.1 软件老化

11.2 老化类别

11.3 度量元模型

11.4 可信计算框架

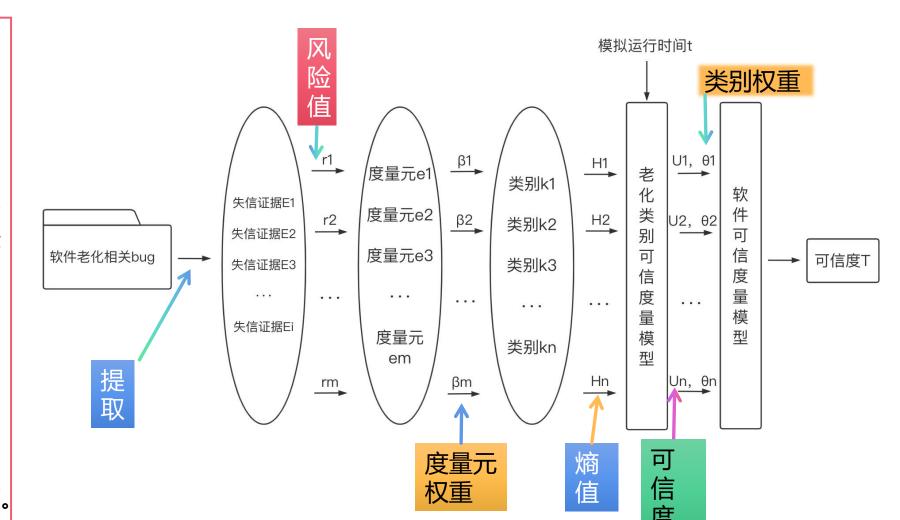
可信计算框架

- 1. 从研究软件老化相关 bug 出发,首先从 bug 中提取失信证据、 设计度量 元,得到了基于软件老化的失信证据度量元。
- 将度量元按导致软件老化的原因分组,分别计算各类别的可信度。
- 3. 将分类可信度进行汇总,得到综合的 软件可信度量模型。
- 4. 以各类别包含的度量元个数以及度量元对应的实例数 为数据, 采用 Brassard 优先次序全面分析标准法 计算权重。
- 5. 输入模拟运行时间,得到预期时间点的软件可信度。
- 6. 从软件老化角度提出 了从数据采集到可信性度量模型的完整框架,如图X.1所示

可信计算框架

从软件老化角度提出 了从数据采集到可信性度量模型的完整框架,如图X.1所示

其中,E1 . . . Ei 表示从搜集到 的软件老化相关 bug 中提取出 的 i 个失信证据, m 表示基于软件老化的失信证 据度量元个数,用 e1...em 表示, r1...rm 表示具体度量 元对应的风险值, β1...βm 表示 具体度量元对应的权重, n 表示 度量元涉及 n 种导致软件老化 的原因, k1...kn 表示度量元 按软件老化的原因分成了 n 个 类别, H1... Hn 表示具体软件 老化原因分类的熵, U1...Un 表示具体类别对应的可信度, θ1...θn 表示具体类别对应的权 重, t表示软件的运行时长, T 表示预测软件在 t 时刻的可信度。



βi 为第 i 个度量元的权重,权重采用统一的计算方式。权重 βi 代表了各度量元与同类型度量元相比的相对重 要程度,数值越大说明该度量元对分类可信度的影响也就越大。这里用一个类别 ARU为例子,每个类别都有其各自对应的一组 βi。

类别ARU共有 2个度量元: A1浮点数寄存器误差(1), A2系统功能异常(3).名称后面的括号内是该度量元包含的实例数目, 例 如"浮点数寄存器误差(1)"代表了这个度量元对应了一份问题报告: 爱国者导弹事件,而系统功能异常(3)代表3个问题报告,分别是 Linux3134, Linux5870, Linux7959。

类别ARU共有 2个度量元: A1浮点数寄存器误差(1), A2系统功能异常(3).名称后面的括号内是该度量元包含的实例数目, 例 如"浮点数寄存器误差(1)"代表了这个度量元对应了一份问题报告: 爱国者导弹事件,而系统功能异常(3)代表3个问题报告,分别是 Linux3134, Linux5870, Linux7959。

这里的数据代表了该度量元与其他度量元重要 程度的比较结果,通过每个度量元在统计数据中出现次数之比反应各度量元重 要性的不同,该度量元对应的错误报告越多,说明该度量元对应的软件老化问题 出现的越频繁,那这个度量元就越重要,对分类可信度的影响也就越大。

构造一个矩阵:

类别ARU度量元权重

	A1(1) 浮点数寄 存器误差	A2(3) 系 统 功 能异常	行总计	(权重βi)
A1 (1)		1/3	1/3	0.10
A2(3)	3/1		3	0.90
列总计	3	1/3	10/3	

 $\beta 1 = (1/3)/(10/3) = 0.10$, $\beta 2 = (3)/(10/3) = 0.90$.

由于度量对象可能是一个复杂的软件,涉及众多可能的软件老化原因和状态参数,因此采用在处理高度复杂问题时效果出众的优先顺序矩阵来解决。 Brassard 确定权重βi的计算方法

类别LOG共有7个度量元:嵌套字功能异常(2)、适配器选择错误(1)、未刷新缓存(1)、死锁(2)、无效更新(1)、临时文件(1)、嵌套字用尽(1)。名称后面的括号内是该度量元包含的实例数目,例如"死锁(2)"代表了死锁这个度量元对应了两份问题报告,分别是 MySQL- Bug52814 和 AXISBug1369。



0.002分配给哪两个度量元? , 取四位小数

Trustworthy Intelligence Group

类别LOG度量元权重

	嵌套字功能 异常 (2)	适配器选择错 误(1)	未刷新缓存(1)	死锁 (2)	无效更新 (1)	临时文件 (1)	嵌套字用尽 (1)	行总计	权重;总计 所占百分比 βi
嵌套字功 能异常 (2)		2(=2/1)	2	1	2	2	2	11	0.234=11/47 0.23404255319
适配器选 择错误 (1)	1/2(=1/2)		1	1/2	1	1	1	5	0.106=5/47 (0.1063829) (0.1064)
未刷新 缓存(1)	1/2	1	2	1/2	1	1	1	5	0.106=5/47 (0.1064)
死锁 (2) 无效更新 (1)	1/2	1	1	1/2	2	1	1	5	0.2340=11/47 0.106=5/47
临时文件	1/2	1	1	1/2	1		1	5	(0.1064) 0.106=5/47 (0.1064)
嵌套字用 尽(1)	1/2	1	1	1/2	1	1		5	0.1064) 0.106=5/47 (0.1064)
列总计	3.5	8	8	3.5	8	8	8	47	0.998(1)



类别LOG度量元权重

类别LOG有7个度量元的权重为:

- 嵌套字功能异常(β1) =0.2340、
- 2. 适配器选择错误 (β2) =0.1064、
- 3. 未刷新缓存 (β3) =0.1064、
- 4. 死锁 (β4) =0.2340、
- 5. 无效更新 (β5) =0.1064、
- 6. 临时文件 (β6) =0.1064、
- 7. 嵌套字用尽 (β7) =0.1064。

类别MEM度量元权重

	M1(2)	M2(7)	M3(1)	M4(1)	M5(3)	M6(3)	M7(3)	M8(1)	M9(1)	M10(2)	M11(2)	M12(1)	M13(1)	M14(4)	M15(7)	行总计	(权重βi)
	内存 需求 过大	资源占用	权限 管理 错误	程序 异常 繁殖	空指针 相关问 题	内存越 界	缓冲区 溢出	缓冲区 溢出	引用 未清 空	循环过 度	异常终 止	回滚机制问题	僵尸进 程过多	高并发 缓存问 题	内存泄漏		
M1 (2)		2/7	2/1	2/1	2/3	2/3	2/3	2/1	2/1	2/2	2/2	2/1	2/1	2/4	2/7	239/14	0.0506
M2(7)	7/2		7/1	7/1	7/3	7/3	7/3	7/1	7/1	7/2	7/2	7/1	7/1	7/4	7/7	249/4	0.1845
M3(1)	1/2	1/7		1/1	1/3	1/3	1/3	1/1	1/1	1/2	1/2	1/1	1/1	1/4	1/7	225/28	0.0238
M4(1)	1/2	1/7	1/1		1/3	1/3	1/3	1/1	1/1	1/2	1/2	1/1	1/1	1/4	1/7	225/28	0.0238
M5(3)	3/2	3/7	3/1	3/1		3/3	1/3	3/1	3/1	3/2	3/2	3/1	3/1	3/4	3/7	731/28	0.0774
M6(3)	3/2	3/7	3/1	3/1	3/3		1/3	3/1	3/1	3/2	3/2	3/1	3/1	3/4	3/7	731/28	0.0774
M7 (3)	3/2	3/7	3/1	3/1	3/3	3/3		3/1	3/1	3/2	3/2	3/1	3/1	3/4	3/7	731/28	0.0774
M8(1)	1/2	1/7	1/1	1/1	1/3	1/3	1/3		1/1	1/2	1/2	1/1	1/1	1/4	1/7	225/28	0.0238
M9(1)	1/2	1/7	1/1	1/1	1/3	1/3	1/3	1/1		1/2	1/2	1/1	1/1	1/4	1/7	225/28	0.0238
M10(2)	2/2	2/7	2/1	2/1	2/3	2/3	2/3	2/1	2/1		2/2	2/1	2/1	2/4	2/7	239/14	0.0506
M11(2)	2/2	2/7	2/1	2/1	2/3	2/3	2/3	2/1	2/1	2/2		2/1	2/1	2/4	2/7	239/14	0.0506
M12(2)	2/2	2/7	2/1	2/1	2/3	2/3	2/3	2/1	2/1	2/2	2/2		2/1	2/4	2/7	225/28	0.0238
M13(1)	1/2	1/7	1/1	1/1	1/3	1/3	1/3	1/1	1/1	1/2	1/2	1/1		1/4	1/7	225/28	0.0238
M14(4)	4/2	4/7	4/1	4/1	4/3	4/3	4/3	4/1	3/1	4/2	4/2	4/1	4/1		4/7	246/7	0.1042
M15(7)	7/2	7/7	7/1	7/1	7/4	7/3	7/3	7/1	7/1	7/2	7/2	7/1	7/1	7/4		249/4	0.1845
列总计																337.39	1

类别STO度量元权重

	S1 (1)	S2(1)	S3 (1)	S4(1)	行总计	(权重βi)
	内磁盘空 间泄露	索引节点爆满	问题	写入失 败		
S1 (1)		1/1	1/1	1/1	3	0.25
	4/4					
S2(1)	1/1		1/1	1/1	3	0.25
S3 (1)	1/1	1/1		1/1	3	0.25
S4(1)	1/1	1/1	1/1		3	0.25
列总计	3	3	3		12	

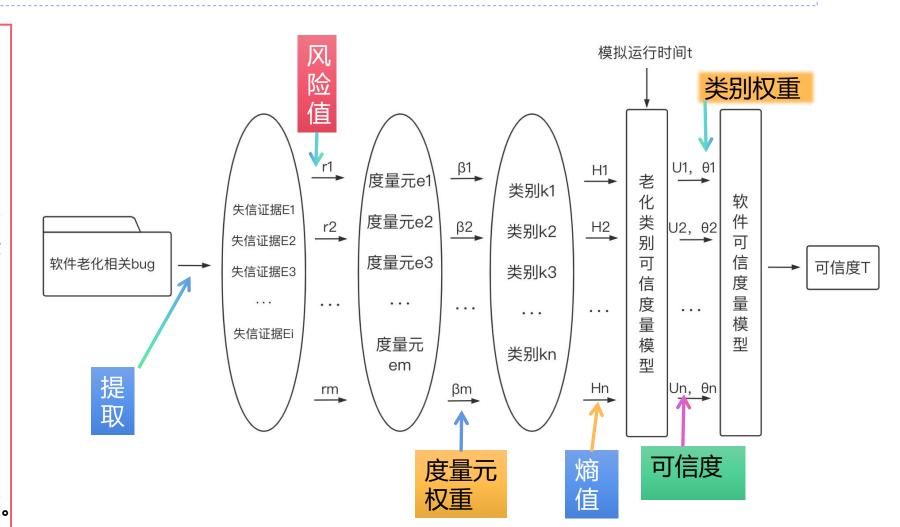
类别NUM度量元权重

	N1 (1)	N2(2)	行总计	(权重βi)
	数值溢出	舍入误		
		差累计		
,				
N1 (1)		1/2	1/2	0.20
N2 (2)	2/1		2	0.80
114 (4)	<i>Z/</i> I		<i>L</i>	0.60
列总计	2	1/2	5/2	

可信计算框架

从软件老化角度提出 了从数据采集到可信性度量模型的完整框架,如图X.1所示

其中,E1 . . . Ei 表示从搜集到 的软件老化相关 bug 中提取出 的 i 个失信证据, m 表示基于软件老化的失信证 据度量元个数,用 e1...em 表示, r1...rm 表示具体度量 元对应的风险值, β1...βm 表示 具体度量元对应的权重, n 表示 度量元涉及 n 种导致软件老化 的原因, k1...kn 表示度量元 按软件老化的原因分成了 n 个 类别,H1 . . . Hn 表示具体软件 老化原因分类的熵, U1...Un 表示具体类别对应的可信度, θ1...θn 表示具体类别对应的权 重, t表示软件的运行时长, T 表示预测软件在 t 时刻的可信度。



度量元熵值计算H

基于软件老化的失信证据<mark>度量元的熵</mark>是指仅在单一基于软件老化的失信证据度量元影响下,表示软件运行状态混乱程度的量。

这里运行状态的混乱程度考虑预期内运行状态和风险状态,因为 预期内运 行状态是固定的,风险状态越多软件运行中偏离预期的 可能就越大,系统就越混乱,熵就会增大。

熵值的计算公式如下:

 $S = \log_{10} r(t)$

其中 r(t) 为度量元对应的最大风险值, 度量元可能有两个风险值。

类别熵值计算H

软件老化原因分类的熵是指在由同一种软件老化原因导致的基于软件老化的失信证据度量元影响下,表示软件运行状态混乱程度的量,是软件运行时间的函数。

软件老化原因类别熵值(在时刻t)的计算公式表示为:

$$H(t) = \sum_{i=1}^{n} \beta_i \log_{10} r_i(t)$$

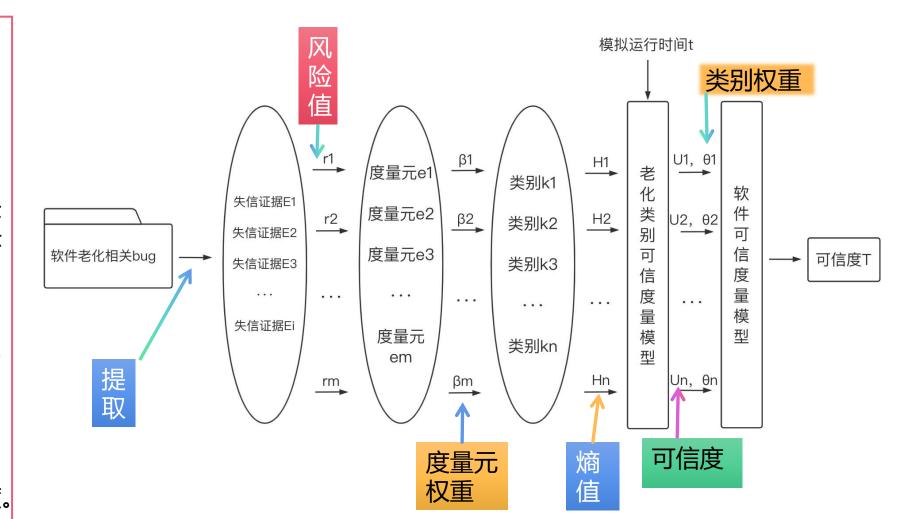
其中 β_i 为第 i 个度量元的权重,权重采用Brassard的计算方式, $r_i(t)$ 为第 i 个度量元的最大风险值。

单一类型的熵值越高,代表了该类型度量元可能引发软件老化的风险越大,软件可信度随时间降低的速度也就越快。

可信计算框架

从软件老化角度提出 了从数据采集到可信性度量模型的完整框架,如图X.1所示

其中,E1 . . . Ei 表示从搜集到 的软件老化相关 bug 中提取出 的 i 个失信证据, m 表示基于软件老化的失信证 据度量元个数,用 e1...em 表示, r1...rm 表示具体度量 元对应的风险值, β1...βm 表示 具体度量元对应的权重, n 表示 度量元涉及 n 种导致软件老化 的原因, k1...kn 表示度量元 按软件老化的原因分成了 n 个 类别,H1 . . . Hn 表示具体软件 老化原因分类的熵, U1...Un 表示具体类别对应的可信度, θ1...θn 表示具体类别对应的权 重, t表示软件的运行时长, T 表示预测软件在 t 时刻的可信度。



软件老化原因类别权重

- 所有涉及的软件老化原因类型,分别为 MEM (15), STO (4), LOG (7), NUM (2), ARU (2)表示五种不同的分类。
- 每个分类后括号内的数字代表了该分类中包 含的度量元数目, MEM (15) 意味着总计有 15 个与内存管理有关的度量元。
- 这里表中的数据代表了该类型与其他类型重要程度的比较结果,通过各分类包含的度量元数目之比反应各分类重要性的不同,包含的度量元数目越多,该类型的软件老化原因相比其他类型越重要,对可信度的影响也就越大。
- 而每一行的行总计即为单个比较得分的总和,括号内是这些总和相 对所有分类的比例权重,也就 是要求的 θi。
- 权重 θi 代表了各软件老化原因的相对重要程度,数值越大说明该 类型的软件老化原因对整体软件可信度的影响也就越大。

软件老化原因类别权重

	MEM (15)	STO (4)	LOG (7)	NUM (2)	ARU (2)	行总计	(权重;
	内存管理	存储空间	逻辑资源	误差累积	未知原因		总计所占百分比 Oi)
MEM (15)		15/4	15/7	15/2	15/2	585/28=	0. 539
						20. 893	
STO (4)	4/15		4/7	2	2	508/105=	0. 125
						4. 838	
LOG (7)	7/15	7/4		7/2	7/2	553/60=	0. 238
						9. 217	
NUM (2)	2/15	1/2	2/7		1	403/210=	0. 049
						1. 919	
ARU (2)	2/15	1/2	2/7	1		403/210=	0. 049
						1. 919	
列总计	1	13/2	23/7	14	14	543/14	1. 000
						38. 786	

类别可信度计算

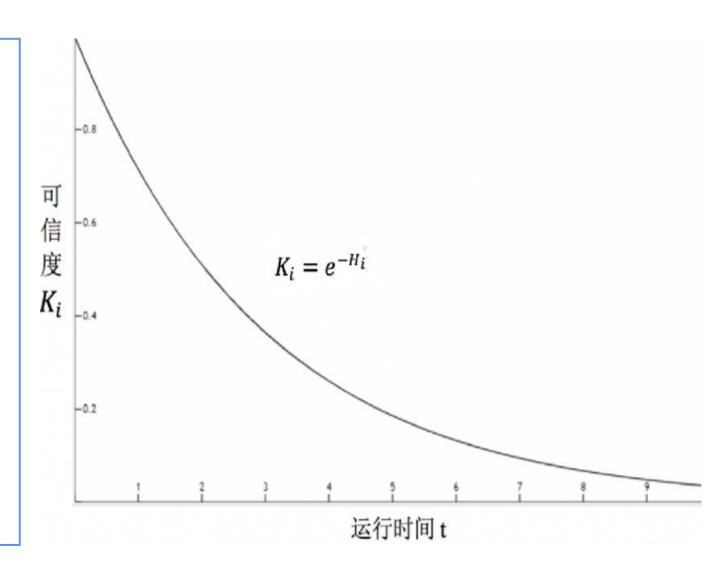
华东师范大学软件学院可信智能团队
TrlG Trustworthy Intelligence Group

设计得到老化类别可信度量模型,各类别可信度与时间 t 的关系函数为:

$$K_i(t) = e^{-H_i(t)}$$

其中, K_i(t) 为第 i 个类别在时刻的可信度, H_i(t) 为第 i 个软件老化类别在时刻的熵, t 为软件运行时间。

可得 K_i(t) 随着 t 的变化曲线大 致如右图所示:



类别可信度计算

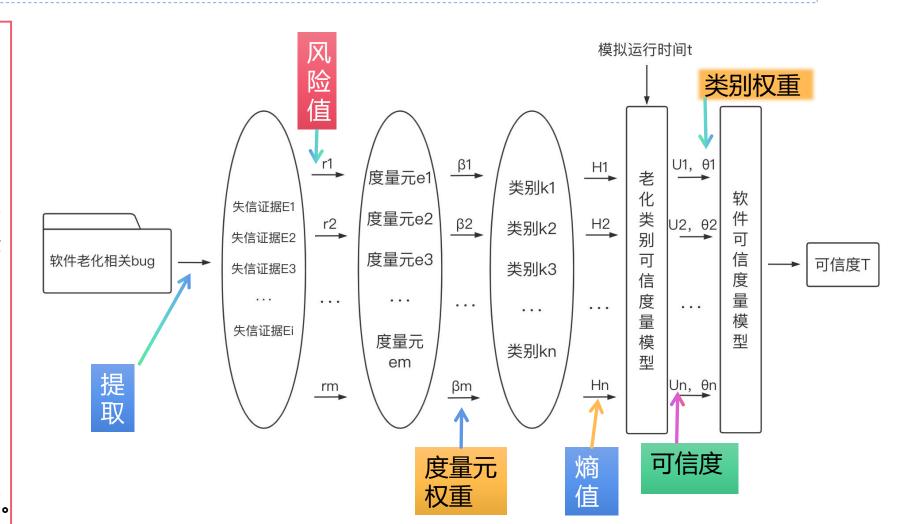
文中提升某一分类的权 重,表示该类型的软件老化原因对软件可信度的影响应增大,因此需要将各类别 (i) 可信度对软件可信度分类可信度的值域 [0, 1] 进行优化,也采用十分制,值域为 [1, 10]。因此i类别可信度的十分制表示为:

$$\mathbf{U}_{i}(t) = \begin{cases} 10e^{-H_{i}(t)}, & 0.1 \le e^{-H_{i}(t)} \le 1\\ 1, & 0 \le e^{-H_{i}(t)} < 0.1 \end{cases}$$

可信计算框架

从软件老化角度提出 了从数据采集到可信性度量模型的完整框架,如图X.1所示

其中,E1 . . . Ei 表示从搜集到 的软件老化相关 bug 中提取出 的 i 个失信证据, m 表示基于软件老化的失信证 据度量元个数,用 e1...em 表示, r1...rm 表示具体度量 元对应的风险值, β1...βm 表示 具体度量元对应的权重, n 表示 度量元涉及 n 种导致软件老化 的原因, k1...kn 表示度量元 按软件老化的原因分成了 n 个 类别,H1 . . . Hn 表示具体软件 老化原因分类的熵, U1...Un 表示具体类别对应的可信度, θ1...θn 表示具体类别对应的权 重, t表示软件的运行时长, T 表示预测软件在 t 时刻的可信度。



软件可信度计算

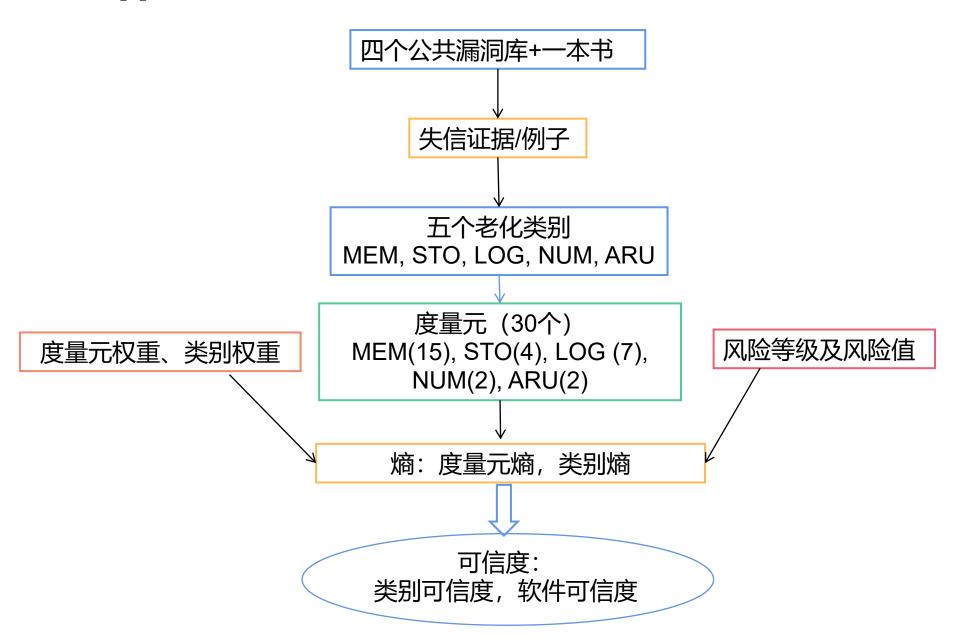
将各分类可信度汇总成综合的软件可信度的软件可信性计算模型如下:

$$T(t) = \begin{cases} \prod_{i=1}^{n} U_i(t)^{\theta_i}, & 1 \le U_i(t) \le 10\\ \sum_{i=1}^{n} \theta_i = 1, & 0 \le \theta_i \le 1 \end{cases}$$

其中, θ_i (i=1, ...,n)为各类别的权重。

总结





例子 (风险值)



华东师范大学软件学院可信智能团队 Trustworthy Intelligence Group

MEM	M1内存需 求过大	M2资源 占用	M3权限管 理错误		M5空指针 相关问题		M7缓冲 区溢出	M8删除表未 释放内存	M9引用 未清空	M10循 环过度	常终	M12回滚 机制问	进程过	M14高并发 缓存问题	M15内 存泄漏
											止	题	多		
权重	0.0506	0.1845	0.0238	0.0238	0.0774	0.0774	0.0774	0.0238	0.0238	0.0506	0.0506	0.0238	0.0238	0.1042	0.1845
t=0	1	1,1	1	1,1	1	1	1	1,1	1	1	1,1	1,1	1,1	1	1
t=10	7	4,7	9	7,4	7	4	7	9,7	7	1	7,7	7,9	1, 1	4	7
ST0	S1磁盘空	区间泄露	S2索引节	点爆满	S3数据插	入问题	S4写入	夫败							
权重	0.2	25	().25	0.2	5		0.25							
t=0	1,	1		1,1	1			1							
t=10	4,7		7,4		7		9								
LOG	L1 嵌套匀	字功能异	L2未刷新	デ 缓存	L3死锁		L4无效	更新	L5临时 时清理	文件未及	. L6 适四错误	记器选择	L7嵌套气	字用尽	
权重	0.2340		0.1064		0.1064		0.2340		0.1064		0.1064	1	0.1064		
t=0	1		1		1		1		1		1		1		
t=10	7		10		7		9		4		10		4		
NUM	N1数值溢	盆出	N2舍入设	是差累计											
权重	0.2		0.8												
t=0	1		1												
t=10	9		7												
ARU	A1浮点数 误差	存存器	A2系统巧	的能异常											
权重	0.10		0.90												
t=0	1		1,1												
t=10	7		7,9												

计算四个类别的熵和可信度,以及软件可信度

	时间	MEM	STO	LOG	NUM	ARU
熵	t=0	0	0	0	0	0
	t=10	0.746	0.872	0.852	0.867	0.943
可信度	t=0	10	10	10	10	10
	t=10	4.743	4.180	4.266	4.202	3.893

时间	软件可信度
t=0	10
t=10	4.482

作业 (风险值)



华东师范大学软件学院可信智能团队 TrlG Trustworthy Intelligence Group

MEM	M1内存需 求过大	M2资源 占用	M3权限管 理错误	M4程序异 常繁殖	M5空指针 相关问题		M7缓冲 区溢出	M8删除表未 释放内存	M9引用 未清空	M10循 环过度	常终	M12回滚 机制问	进程过	M14高并发 缓存问题	M15内 存泄漏
											止	题	多		
权重	0.0506	0.1845	0.0238	0.0238	0.0774	0.0774	0.0774	0.0238	0.0238	0.0506	0.0506	0.0238	0.0238	0.1042	0.1845
t=0	4	4,1	1	1,4	1	1	4	1,1	1	1	4,1	1,1	1,4	1	4
t=10	4	4,7	7	4,4	9	4	7	4,7	7	4	7,7	7,4	7, 4	9	7
ST0	S1磁盘空间泄露		S2索引节点爆满		S3数据插入问题		S4写入失败								
权重	0.25		0.25		0.25		0.25								
t=0	1,4		1,1		4		4								
t=10	4,7 7,9			7		7									
LOG	L1 嵌套字功能异常		L2未刷新	下缓存	存 L3死锁		L4无效更新		L5临时文件未及 L6适配器选择 时清理 错误			L7嵌套字用尽			
权重	0.2340		0.1064		0.1064		0.2340		0.1064		0.1064	0.1064		0.1064	
t=0	1		1		4		1		1		1		7		
t=10	7		10		7		4		7 10		10		9		
NUM	N1数值溢出		N2舍入误差累计												
权重	0.2 0.8														
t=0	4 1														
t=10	7 4														
ARU	A1浮点数寄存器 A2系统功能异常 误差														
权重	0.10		0.90												
t=0	1 1,4														
t=10	7		7,7												

作业:编程或使用课程提供的Python Trie Trustworthy Intelligence Group



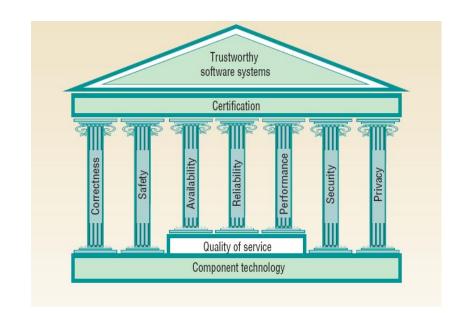
输入: 类别, 以及类别中每个度量元的含例子个数,

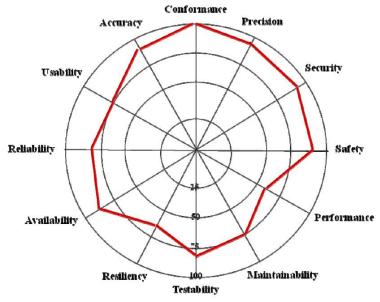
输出:在时刻t=0和t=10情况下,四个类别的熵和可信度,以

及软件可信度。









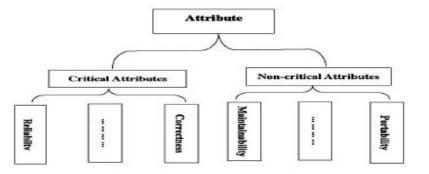


Fig. 1. The Hierarchical Model for Trustworthy Metrics of Software



度量计算模型

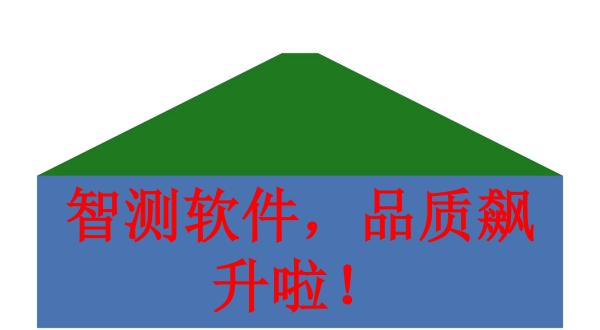
$$\begin{cases} \alpha_1 + \alpha_2 + \dots + \alpha_n = \sum_{i=1}^n \alpha_i = 1 & 0 \le \alpha_i \le 1 \\ T = y_1^{\alpha_1} y_2^{\alpha_2} \dots + y_n^{\alpha_n} = \prod_{i=1}^n y_i^{\alpha_i} & 0 \le y_i \le 1 \end{cases}$$

T为软件可信度, $0 \le T \le 1$ yi为第i个属性的可信度 α_i 为第i个属性的权重值

每个可信属性的分布函数是什么?

嵌入式软件在运行10个小时后, 其可信等级为IV的概率是多少?

制定嵌入式软件可信评估行业标准与增强指南,推动嵌入式软件的可信开发与应用。





谢谢