



组合云系统可靠性概念漂移在线度量方法研究

王磊^{1*}, 张云秋¹, 徐丙凤², 徐逸卿²

1. 南京林业大学管理科学与工程系, 南京 210037

2. 南京林业大学软件工程系, 南京 210037

* 通信作者. E-mail: leiwang@njfu.edu.cn

收稿日期: 2020-05-03; 修回日期: 2020-07-01; 接受日期: 2020-09-04; 网络出版日期: 2021-09-13

江苏省六大人才高峰高层次人才培养项目 (批准号: RJFW-029) 和教育部人文社科基金一般项目 (批准号: 18YJCZH170) 资助

摘要 随着全球云计算产业规模的迅速增长, 采用按需付费的方式, 基于云服务组合实现大规模软件服务的动态集成和协同, 从而构造复杂软件, 已成为一种切实可行的方法. 所构建的组合云系统运行在动态、不确定的环境下, 如何应对系统运行可靠性数据流的概念漂移问题, 保障系统运行质量, 是一个亟待解决的挑战性问题. 为给针对组合云系统的可靠性自适应提供早期指导, 本文基于主动式缺陷管理思想, 研究组合云系统运行风险的在线感知方法, 提出基于 Sinkhorn 距离的可靠性概念漂移在线度量方法 RCDMeas. 这一方法通过引入熵正则化和不动点迭代, 计算系统历史累积的可靠性流数据和临近的可靠性流数据的分布距离, 从而识别组合云系统可靠性概念漂移及系统潜在运行风险. 大规模数据实验结果验证了本文方法的有效性. 本文的工作对服务组合系统的自动化运维具有重要的理论意义与实际应用价值.

关键词 组合云系统, 可靠性, 概念漂移, 在线, 运行质量保障

1 引言

近年来我国在金融、交通、制造等领域对能够支持需求动态不确定性、具有智能架构的大规模复杂软件系统的应用需求逐渐凸显. 作为构建大规模复杂软件系统的一种重要思想, 服务组合技术逐渐引起学术界及工业领域有关人员的高度关注^[1,2]. 随着全球云计算产业规模的迅速增长, 大量公开可用的云服务应用程序接口 (application program interface, API) 不断推出市场, 这为服务组合技术的工业化应用带来前所未有的机遇^[3]. 采用按需付费的方式租用云服务 API, 开发人员可以通过服务组合技术快速地开发并部署一个满足特定用户需求的组合云系统, 以满足对大规模复杂软件系统的应用需求^[4,5].

引用格式: 王磊, 张云秋, 徐丙凤, 等. 组合云系统可靠性概念漂移在线度量方法研究. 中国科学: 信息科学, 2021, 51: 1438–1450, doi: 10.1360/SSI-2020-0117
Wang L, Zhang Y Q, Xu B F, et al. Reliability concept drift online measurement for composite cloud systems (in Chinese). Sci Sin Inform, 2021, 51: 1438–1450, doi: 10.1360/SSI-2020-0117

然而, 所构建的组合云系统运行在动态多变的网络环境下, 每一个组件系统的负载会出现不确定性波动. 组件系统多变的外部运行环境以及不确定的内部工作状态 (包括支撑组件系统运行的物理机器实时的 CPU 使用率、磁盘 I/O、内存使用情况、函数/进程的调用等), 将会使得组件系统面临潜在的软件、硬件错误, 以及性能异常^[1,6]. 单个组件系统的错误甚至还可能会导致级联效应, 使得整个系统无法正确运行^[7]. 面向多样的云系统用户应用需求, 组合云系统的组合结构会随时发生改变, 这会导致组合系统中组件系统间的错误依赖关系及级联效应千变万化^[8,9].

以上因素, 使得组合云系统的可靠性会随时间发生不确定性波动, 其可靠性数据流的分布规律相对于历史的观测记录而言会随时发生变化. 这种变化可能是季节性的, 也可能是难以预计的突变或剧变. 特别地, 在动态变化的环境下, 组合云系统不断产生的可靠性数据流, 其分布特征会随时发生不确定性的变化, 这种数据流分布特征的改变被定义为可靠性概念漂移^[10~12]. 因为针对组合云系统可靠性的历史观测数据很难全面覆盖可靠性概念漂移场景下的样本数据, 这使得现有的基于收集历史的观测数据而构建的系统容错、可靠性预测等模型^[8,13~15] 在组合云系统运行过程中的性能 (准确性、可扩展性等) 会出现波动, 导致在组合云系统中部署的性能保障机制失去效果. 这使得组合云系统的运行维护工作面临潜在的可靠性概念漂移问题^[13]. 不同于一般的企业内部集成云系统, 组合云系统需要能够实时地感知其可靠性的概念漂移, 在线地自适应更新其容错架构, 以提高系统应对不确定性运行风险的能力. 由此看来, 如何实时感知组合云系统运行可靠性的概念漂移, 为开展系统可靠性自适应提供早期指导, 进而保证组合云系统能够满足应用的服务等级协议 (service-level agreements, SLA) 约束, 持续地提供稳固服务, 这是组合云系统研究和应用领域一个新的亟待解决的挑战性问题.

不同于以往的研究, 为应对组合云系统运行可靠性的概念漂移问题, 降低系统失效风险, 保障系统可以持续、健康地运行, 本文拟基于主动式缺陷管理思想^[16], 通过针对服务性能参数、系统架构信息等大数据的分析, 实时感知组合云系统的可靠性概念漂移及其揭示的系统潜在运行风险, 为面向组合云系统的系统可靠性自适应应用提供实时参考.

由于组合云系统中各个组件系统内部工作状态以及外部运行环境的不确定性变化, 其可靠性概念漂移的发生不具有明显的时间规律. 发生在不同组件上不同程度的可靠性概念漂移对组合云系统运行质量的威胁不同. 有些漂移对组合云系统可靠性的威胁较小, 有些漂移可能会导致组合云系统可靠性发生明显的衰减, 使得系统无法满足 SLA 的应用需求. 为了能够准确判断可靠性概念漂移对组合云系统的危害程度, 从而决定是否有必要启动组合云系统自适应机制, 本文中我们需要能够定量地度量组合云系统的可靠性概念漂移的程度及其揭示的系统运行风险. 面向一个云服务组合系统, 开展针对可靠性概念漂移的定量度量, 这对传统的面向流数据的概念漂移检测方法提出新的挑战^[10].

本文首先根据系统组合结构以及各个组件系统的可靠性值, 采用聚合函数及递归函数分别计算不同结构的组合云系统可靠性. 其次, 基于 Sinkhorn 距离, 采用熵正则化和不动点迭代方法, 计算组合云系统累积的历史可靠性数据流与临近时间段的可靠性数据流之间的分布距离. 使用距离计算结果作为组合云系统可靠性概念漂移的定量度量. 再次, 设计可靠性概念漂移风险度量函数, 定量地计算可靠性概念漂移的风险. 在此基础上, 提出一种基于 Sinkhorn 距离的组合云系统可靠性概念漂移在线度量方法 (reliability concept drift measure via Sinkhorn distance for composite cloud systems, RCDMeas). 基于度量结果, 如果组合云系统可靠性的负向概念漂移大于某一阈值, 则触发预警机制. 本文的工作对组合云系统自动化运维应用具有重要的理论意义与实际应用价值. 为基于服务组合的云系统运行质量保障提供一种有效的解决方案, 对推动服务组合系统的工业化应用具有重要意义.

2 组合云系统可靠性

在介绍可靠性概念漂移度量方法之前, 我们首先给出本文中针对组合云系统可靠性的度量方法. 不同的学科, 面向不同的应用, 有关可靠性的定义不尽相同. IEEE 标准 ANSI/IEEE STD-610.12-1990 给出了有关软件可靠性的定义: 软件可靠性是指在一个特定的环境下, 一个特定长度的时间段内, 软件系统能够始终保持无故障运行的概率^[17]. 这一定义被广泛地应用于传统的软件系统、Web 服务等领域. 借助服务组合技术所构建的组合云系统是一种 SaaS (software as a service) 应用, 自然也适用于这一定义.

常见的用于评价系统可靠性的指标主要有: (1) 需求失效率 (probability of failure on demand, PoFoD); (2) 错误发生率 (rate of failure occurrence, RoCoF); (3) 平均出错时间 (mean time to failure, MTTF); (4) 平均错误间隔时间 (mean time between failures, MTBF); (5) 危机率 (hazard rate), 即系统能够维持正确地运行, 直到 t 时刻; (6) 错误率或错误概率 (failure-rate or failure probability) 等. 这些指标分别被用于描述基于不同应用场景的系统的可靠性. 一般而言, 传统的有关复杂系统可靠性的研究, 主要考虑的是两类可靠性模型, 即可靠性衰减模型和可靠性增长模型^[18]. 其中, 可靠性衰减模型主要适用于系统性能随着时间逐渐衰退的系统, 如机械系统逐渐受到磨损, 性能逐渐衰减. 可靠性增长模型在传统的计算机系统 (特别是软件系统) 运行质量保障的研究中较为常见, 例如, 软件系统的 BUG 不断得到修复, 其系统的可靠性会不断提高.

与上述系统不同的是, 在组合云系统中, 组件系统的内部工作状态、外部运行环境, 以及组件系统的负载量等因素的实时变化具有动态、不确定性, 组件系统出现错误的响应 (或性能异常) 的概率分布不会表现出明显的时间规律. 这使得 RoCoF, MTTF, MTBF, hazard rate 等指标均不适用于描述组合云系统中组件系统的可靠性. 在一个组合云系统中, 每一个组件并不是被长期调用的, 系统会根据实时的执行过程, 动态地调用每一个组件系统. 我们可以使用一个伯努利 (Bernoulli) 测试过程来描述每一次系统调用的结果, 即系统的调用结果为成功或失败 (1 或 0). 参照文献 [19] 的做法, PoFoD 可以被用于评估组合云系统的可靠性. 事实上, 相似的方法早已被用于计算面向服务的系统的可靠性, 如文献 [14] 通过在固定长度时间段内对组件系统开展连续地调用测试, 以此来评估组件系统在相应时间段内的错误率, 并据此计算组件系统的可靠性.

不同于传统的计算机系统, 组合云系统中的每一个组件系统, 其性能会随时间发生波动, 在对组件系统进行调用测试的过程中, 组件系统的性能异常 (如长期得不到响应或响应超时) 同样会使得组件系统不满足用户的需求. 为使得组件系统调用能够在期望的时间内获得正确的响应, 我们定义以下两类情况为组件系统的调用错误: (1) 返回错误的响应; (2) 性能异常 (如超时)^[15]. 因此, 本文对 PoFoD 定义作进一步改进, 即将性能异常视为一种系统调用错误, 给出以下性能敏感的需求失效率评价指标.

定义1 (性能敏感的需求失效率 (performance-aware probability of failure on-demand, paPoFod)) 假设在一个固定的时间段 Δt 内 (时间长度为 $\text{len}(\Delta t)$), 我们每隔 s 秒向一个组件系统发送一个调用请求 (共发送 n 次), 每一次调用请求为一次伯努利测试, 即结果为 $L = 0/1$. 假设用户能够容忍的最大的响应时间约束为 RT_{\max} . paPoFod 定义为在组件系统调用过程中, 在 RT_{\max} 约束的条件下, 不成功的响应的概率, 即 $f(\Delta t) = \Pr(L = 0)$.

如前所述, 根据 IEEE ANSI/IEEE STD-610.12-1990 标准, 可靠性描述的是软件系统在残存时间 Δt 结束前能够始终保证无故障运行的概率. 根据定义 1, 由于在 Δt 时间段内共进行了 n 次伯努利测试, 这 n 次测试始终保持错误的率值 (即组件系统在 Δt 时间段内任意时刻出错的概率, 一般表示为

表 1 组合云系统需求失效率聚合函数

Table 1 Aggregation functions of paPoFod for composition cloud systems

Composite structure	Aggregation function
Sequence or parallel	$\lambda_{\Phi} = 1 - \prod_{i=1}^n (1 - \lambda_i)$
Branch	$\lambda_{\Phi} = 1 - \sum_{i=1}^n b_i(1 - \lambda_i)$
Loop	$\lambda_{\Phi} = 1 - \sum_{i=0}^n l_i(1 - \lambda_1)^i$

错误率 λ) 为

$$\lambda = 1 - [1 - f(\Delta t)]^n. \quad (1)$$

基于指数可靠性表达式, 我们通过以下函数计算一个组件系统 i 的可靠性:

$$r_i^{\Delta t} = \exp[-\lambda \times \text{len}(\Delta t)]. \quad (2)$$

值得注意的是, 指数可靠性表达式是一种常用的度量服务系统可靠性的方法^[14]. 其物理意义在于, 通过统计组件系统在单位时间长度内出现错误的率值 (其中, 率值是指需要考虑残存时间的一种概率值), 来推导组件系统的存活率 (即可靠性). 在此, 我们给出其计算依据. 设在 Δt 时间段内存在 n 个等间隔的时间点 t_1, t_2, \dots, t_n , $f(\Delta t)$ 表示组件系统在 t_n 时刻的失效概率, $r_i^{\Delta t}$ 表示组件系统在 t_n 时刻的存活率 (直到 t_n 时刻不出错的概率, 也即 Δt 时间段的可靠性), 则有 $f(\Delta t) = F'(\Delta t)$, 及 $r_i^{\Delta t} = 1 - F(\Delta t)$. 其中, $F(\Delta t)$ 表示组件系统在 Δt 时间段内的失效率, 则有

$$\lambda = \frac{f(\Delta t)}{r_i^{\Delta t}} = \frac{F'(\Delta t)}{1 - F(\Delta t)}. \quad (3)$$

解此微分方程可得到一个特解 $F(\Delta t) = 1 - \exp[-\lambda \times \text{len}(\Delta t)]$. 据此即可得到式 (2). 其中, $\text{len}(\Delta t)$ 表示 Δt 时间段的时间长度.

对于一个顺序、并行、分支、循环结构的组合系统 Φ 而言, 其需求失效率 λ_{Φ} 可以通过基于组合系统 workflow 结构分析的聚合函数计算 (表 1). 其中, λ_i 表示第 i 个分支的错误率, b_i 表示第 i 个分支被执行的概率, l_i 表示第 i 个分支被执行的次数.

对于服务组合系统容错应用而言, 还有一类较为常见的结构, 即 NVP (N-version programming) 容错结构. 我们将 NVP 容错结构建模为一种 k -out-of- n 系统, 即一个 NVP 容错系统 Φ 是正确的, 当且仅当在 n 个组件系统中, 至少有 k 个组件能够返回正确的结果. 对于一个 NVP 容错系统结构而言, 因为每个组件系统的可靠性是不同的, 我们很难借助表 1 中简单的排列组合的方法设计聚合函数, 我们采用以下递归函数来计算一个 k -out-of- n 系统的错误率^[20]:

$$\begin{aligned} \lambda_{\Phi} &= \lambda_{k/n} = \lambda_n \times \lambda_{k/n-1} + (1 - \lambda_n) \times \lambda_{k-1/n-1}, \\ \lambda_{0/n} &= 0, \\ \lambda_{j/i} &= 1, \text{ when } j > i, \end{aligned} \quad (4)$$

其中, $\lambda_{k/n}$ 表示 k -out-of- n 系统的错误概率, λ_n 表示第 n 个组件的错误概率, $\lambda_{k/n-1}$ 表示在第 n 个组件出错的情况下, 组合系统的条件错误概率.

至此, 通过伯努利测试的方法, 可以依据式 (2) 计算每个组件系统在一个特定时间段内的可靠性值. 若已知每个组件系统的错误率值, 再根据表 1 及式 (4) 的计算方法得到不同组合结构的组合云系统的错误率值, 进而得到组合云系统的可靠性值.

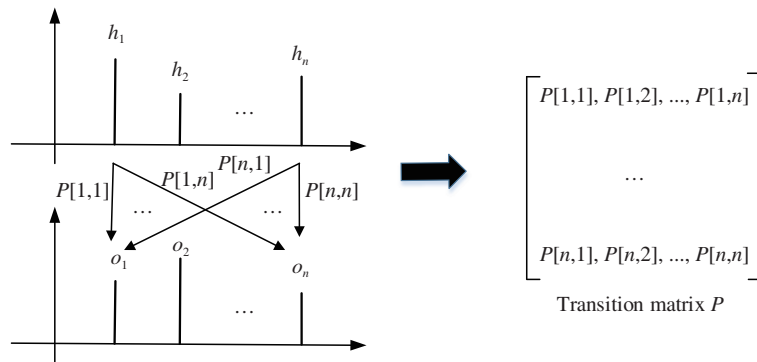


图 1 最优运输问题

Figure 1 Optimal transport problem

3 RCDMeas 方法

基于以上计算, 将能够得到组合云系统的可靠性流数据, 记为 $r_{\Phi}^{\Delta t}(1), r_{\Phi}^{\Delta t}(2), \dots$. 根据式 (2), 每个可靠性值属于 $[0,1]$ 区间. 以下将介绍针对组合云系统的可靠性概念漂移在线度量方法 RCDMeas.

设在组合云系统累积的历史可靠性数据流和临近的可靠性数据流中, 依据可靠性值的区间 $[0, 0.1), [0.1, 0.2), \dots, [0.9, 1]$ 开展统计, 其概率分布列向量分别为

$$\begin{aligned}
 h &= \left\{ h_i \in \mathbb{R}_+^n : \sum_{i=1}^n h_i = 1 \right\}, \\
 o &= \left\{ o_i \in \mathbb{R}_+^n : \sum_{i=1}^n o_i = 1 \right\},
 \end{aligned} \tag{5}$$

其中 i 为可靠性值的区间对应的编号, n 为总的区间个数, h 为累积的历史可靠性数据流分布列向量, o 为临近的可靠性数据流分布列向量.

面向组合云系统可靠性概念漂移的度量, 这里需要解决的是两个问题: 首先, 在线检测组合云系统的可靠性是否发生概念漂移; 其次, 分析可靠性概念漂移对组合云系统的潜在威胁. 为使得可靠性概念漂移度量结果便于比较, 我们需要能够定量地对概念漂移的程度 (即分析可靠性流数据分布规律发生了多大的改变) 和其对组合云系统造成的潜在威胁程度 (即负向可靠性概念漂移的程度) 进行评估. 有些可靠性的概念漂移表明组合云系统总体可靠性变得更好 (即在临近的系统可靠性数据流中, 可靠性值更高的样本分布量更多), 有些表明系统总体可靠性变差, 这对可靠性概念漂移的度量问题带来挑战. 传统的基于差异检测的概念漂移检测方法 (如基于 KL 散度的概念漂移检测方法), 因其非对称性评估结果, 很难适用.

Sinkhorn 算法通过寻求两组数据流分布的最小转换成本, 来度量两组分布的距离, 具有严格的数学理论基础. 本文基于 Sinkhorn 算法, 提出 RCDMeas 方法, 计算两组可靠性数据流分布的差异, 进而度量组件系统的可靠性概念漂移. 具体而言, Sinkhorn 算法所要解决的问题是一个最优运输问题, 即将 h 转变为 o (假设 h, o 表示土堆的分布向量), 所需要花费的最低运输成本, 这一成本定义为 Sinkhorn 距离^[21, 22]. 如图 1 所示, 为了将 h 转变为 o , 我们需要将 h 中的每一个分量转换到 o 的每一个分量中. 转变矩阵 P 表示一种可能的转变方案. 这种转变方案是不唯一的.

为了表示所有可能的转变方案, 本文定义一个转变多面体:

$$U(h, o) = \{P \in \mathbb{R}_+^{n \times n} | P1_n = h, P^T 1_n = o\}, \quad (6)$$

其中 P 为转变矩阵, $P[i, j]$ 表示如果将 h 转变为 o , 需要将 h 中的第 i 组分布移入 o 中的第 j 组分布的数量. 1_n 是一个 n 维值全部为 1 的列向量.

在进行 $P[i, j]$ 转变的过程中, 存在一个确定的成本值. 例如, 对于最优运输问题而言, 这表示将 h_i 数量的土移入 o_j 所要花费的运输成本. 为此, 我们定义转变成本矩阵 M , $M[i, j]$ 表示将单位数量的可靠性值分布从历史的可靠性分布的列向量的第 i 组转变为临近时间段的可靠性分布的列向量的第 j 组, 所要花费的成本. 为了保证转变成本的一致性, 令

$$M[i, j] = \begin{cases} |i - j| \times \delta, & i \neq j, \\ 0, & i = j, \end{cases} \quad (7)$$

其中, δ 为一个固定的常量.

至此, 我们可以通过寻求 h 到 o 的最低运输总成本来定量地度量 h 到 o 的概念漂移:

$$L(h, o) = \min \langle P, M \rangle = \langle P^\xi, M \rangle, \quad (8)$$

其中, $\langle \cdot, \cdot \rangle$ 为 Frobenius 点积运算. P^ξ 可以通过熵正则化方法计算:

$$P^\xi = \arg \min_{P \in U(h, u)} \langle P, M \rangle - \xi \times H(P), \quad (9)$$

其中,

$$H(P) = - \sum_{i, j \in \{1, 2, \dots, n\}} P[i, j] \log P[i, j], \quad (10)$$

其中, ξ 为拉格朗日乘子, 这里的 P^ξ 是一个近似最优解, 即近似最优的转变矩阵.

借助 Sinkhorn 矩阵缩放算法, 有

$$P^\xi = \text{diag}(u)K\text{diag}(v), \quad (11)$$

其中, K 为核矩阵. 通过不动点迭代, 不断更新 u, v 的值, 算法将最终收敛于 P^ξ . 其中, 不动点为

$$u = \frac{h}{K \frac{o}{K^T u}}, \quad (12)$$

其中,

$$K = \exp\left(-\frac{1}{\xi}M\right), \quad (13)$$

$$u = \frac{h}{Kv}, \quad (14)$$

$$v = \frac{o}{K^T u}. \quad (15)$$

具体而言, 基于熵正则化和不动点迭代, 求解 h 到 o 的近似最优的转变矩阵 P^ξ 的算法 (reliability concept drift detection via Sinkhorn distance, RCD-SD) 如算法 1 所示.

Algorithm 1 RCD-SD

Input: $M \in \mathbb{R}_+^{n \times n}$, ξ , h , o , δ , MaxIteration.

Primary iteration: P^ξ .

- 1: Initialize K by solving Eq. (13);
 - 2: Initialize $u = [1/n, 1/n, \dots, 1/n]_n$;
 - 3: **while** $d \leq \text{MaxIteration}$ and u changes **do**
 - 4: Update u by solving Eq. (12);
 - 5: $d++$;
 - 6: **end while**
 - 7: Calculate v by solving Eq. (15);
 - 8: Calculate P^ξ by solving Eq. (11);
 - 9: Return P^ξ .
-

至此, 将 P^ξ 代入式 (8), 可以得到概念漂移的度量值 $L(h, o)$. 进而根据 $L(h, o)$ 值的大小判断概念漂移的程度. 比如可以设定阈值 κ , 如果 $L(h, o) \geq \kappa$, 则认为组合云系统发生了概念漂移. 我们有必要进一步分析其风险, 并判断是否有必要启动系统自适应机制.

为了进一步分析概念漂移所揭示的组合云系统运行风险, 我们设计一个针对组合云系统的可靠性概念漂移的风险检测函数. 通过比较式 (5) 中 h 和 o 高于 SLA 中对系统可靠性的要求的那些分布的变化情况来识别可靠性风险. 设 $[r_i]$ 表示第 i 个可靠性值区间的下界 (例如 $[r_3] = 0.2$), 我们将低于 SLA 对可靠性需求的 $[r_i]$ 置为 0:

$$[r_i] = \begin{cases} [r_i], & [r_i] \geq r_{\text{SLA}}, \\ 0, & [r_i] < r_{\text{SLA}}. \end{cases} \quad (16)$$

我们采用以下公式计算可靠性概念漂移的风险:

$$F = \sum_i [r_i] \times (o_i - h_i). \quad (17)$$

当 $F < 0$ 时, 表明组合云系统发生了负向可靠性概念漂移, 此时 $|F|$ 越大, 表明负向可靠性概念漂移越明显, 系统运行风险越大. 当满足以下公式时, 则认为组合云系统发生了严重的负向可靠性概念漂移, 需要触发相应的自适应机制.

$$|F| \geq W, \text{ and } F < 0, W \in [0, 1]. \quad (18)$$

在研究过程中, 我们将根据实验分析, 进一步研究概念漂移风险度量、系统自适应的触发机制, 即如何设定适当的阈值. 拟通过比对不同阈值水平下误报率和漏报率来设定.

4 实验分析

本文通过开展大规模数据实验验证 RCDMeas 方法的有效性. 为便于验证可靠性概念漂移度量结果的有效性, 本文定义一种基准的概念漂移检测方法, 即定义在某一可靠性数据流中, 当组合云系统的可靠性的分布低于 SLA 约束达到 20% 以上时, 即认定为组合云系统发生了负向可靠性概念漂移. 本文将分别利用 RCDMeas 以及其他主流的概念漂移检测方法分别检测可靠性流数据的概念漂移, 并将检测结果与基准概念漂移检测方法的检测结果进行比较, 通过分析不同方法的准确率、精确率、命中率、F1 分数等指标, 比较不同方法的有效性.

4.1 实验设置

在实验过程中,首先使用大规模网络分析工具包 Pajek 来模拟服务组合 workflow. 通过生成随机有向图来表示云服务组合 workflow 结构,图中的每一个顶点表示一个组件云服务,每个有向弧表示服务的调用. 其次,对于 workflow 中的每一个云服务的性能数据,本文分别使用 QWS v2 数据集^[23]和 ICWS 2012 大规模真实服务性能数据^[24],开展模拟的数据实验. 其中,针对 QWS v2 数据集,我们随机地为 workflow 中的每个节点分配一个 QWS 中的服务. 在 QWS v2 中包含 2507 个真实的服务性能数据. 为了模拟可靠性数据流,本文为每个服务 i 生成一个介于 $[0,1]$ 的随机数 rand_i . 在此基础上连续生成介于 $[r_i \times \text{rand}_i, (100\% - r_i) \times \text{rand}_i + r_i]$ 的随机数,模拟其历史累积的可靠性数据流,其中 r_i 表示 QWS 中服务 i 的可靠性值 ($[33\%, 89\%]$). 进一步地,为模拟可靠性的概念漂移,我们在生成数据流的过程中随机改变可靠性值的生成规律,随机生成 $[0\%, 100\%]$ 的可靠性值,从而得到针对每一个组件云服务的临近的时间段的可靠性数据流. 类似地,针对 ICWS2012 数据,本文基于这一数据集中的 QoSwdl, QoSadl, QoSstdl 数据,计算每一个服务的可靠性. 这一数据集中包含了 1770 个 RESTful 和 SOAP 等不同类型服务的连续的响应时间 (response time)、吞吐量 (throughput)、成功率 (successability) 等性能参数. 针对每一个服务而言,至少包含了 28 周连续采集的服务成功率数据 (28 slots). 本文设定 $\text{RT}_{\max} = 1000 \text{ ms}$,进而根据 4 个连续的时间点 (slots) 的响应时间、成功率,通过式 (2) 计算得到一个可靠性值. 在此基础上,采用与 QWS v2 类似的方法,分别得到每个服务的历史累积的可靠性数据流和临近的可靠性数据流. 进一步根据服务组合 workflow,根据表 1 及式 (4) 中的计算公式,得到组合云系统可靠性数据流. 本实验中,取可靠性数据流中最后 50 个时间点的流数据作为临近的可靠性数据流,在此之前的 1000 个时间点的流数据作为历史的可靠性数据流. 在实际的应用中,历史的可靠性数据流的时间窗口长度应尽可能长,临近的可靠性数据流的时间窗口长度应根据应用的实际需求设定. 基于以上数据,本文分别采用不同的方法开展概念漂移及漂移风险检测,并与基准方法进行比较,分析不同方法的有效性、性能. 针对每一种方法,本文分别开展 20 次检测,并计算其平均性能. 本文的实验基于 JupyterLab 0.35.4 平台,采用 Python 语言编程实现. 系统软、硬件平台为: Windows 7 x86 Enterprise Edition OS, Intel(R) Core(TM) i7 2600 CPU, 12 GB RAM, Seagate 1TB HDD.

4.2 比较的方法

现有的概念漂移检测方法主要包括基于错误率的概念漂移检测、基于数据分布的概念漂移检测及基于多假设检验的概念漂移检测方法^[10]. 为验证本文方法的有效性,本文基于文献中的各类概念漂移检测方法另外设计了 3 种方法来开展可靠性概念漂移风险检测,并与本文的方法开展比较. 这 3 种方法主要包括:

- 基于背离率差异的方法 (deviation rate difference based approach, DRD). 我们通过统计历史累积的可靠性数据流与临近的可靠性数据流中背离用户的 SLA 需求的比例的均方根误差来开展概念漂移检测. 设历史累积的数据流的背离率为 $d(h)$,临近的可靠性数据流的背离率为 $d(o)$,则

$$|F| = \sqrt{\frac{(d(h) - d(o))^2}{2}}. \quad (19)$$

- 基于 KL 散度的方法 (Kullback-Leibler divergence based approach, KLD). 我们统计历史累积的可靠性数据流与临近的可靠性数据流分布的相对熵来开展概念漂移检测:

$$L(h, o) = \text{KL}(h||o) = \sum_i P(h_i) \log \frac{P(h_i)}{P(o_i)}, \quad (20)$$

其中 $P(h_i)$ 和 $P(o_i)$ 分别表示在 h 和 o 中 h_i 和 o_i 的概率分布. 进一步再根据式 (18) 判断是否有必要启动自适应机制.

• 基于多重假设检验的方法 (multiple hypothesis test based approach, MHT). 本文采用 Kolmogorov-Smirnov (KS) 检验的基本思想, 通过观察历史累积的可靠性数据流与临近的可靠性数据流累积分段函数的差异来开展概念漂移检测, 即

$$L(h, o) = \sup \frac{\sum_i |h_i - o_i|}{n}. \quad (21)$$

同样地, 本文进一步根据式 (18) 判断是否有必要启动自适应机制.

4.3 性能评价指标

设总的检测次数为 N , 正确地检出 (true positive detection) 为严重的负向可靠性概念漂移的次数为 N_{TPD} , 正确的检测 (true detection) 次数为 N_{TD} , 组合云系统发生严重的负向可靠性概念漂移的总次数为 N_{CD} . 我们设计以下评价指标以度量可靠性概念漂移风险检测的有效性.

- 准确率 (Accuracy):

$$\text{Accuracy} = \frac{N_{TD}}{N} \times 100\%. \quad (22)$$

- 精确率 (Precision):

$$\text{Precision} = \frac{N_{TPD}}{N_{TD}} \times 100\%. \quad (23)$$

- 命中率 (Hit rates):

$$\text{Hit rates} = \frac{N_{TPD}}{N_{CD}} \times 100\%. \quad (24)$$

这一指标检测概念漂移检测方法的敏感性, 因此又称为召回率.

- F1-score:

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Hit rates}}{\text{Precision} + \text{Hit rates}} \times 100\%. \quad (25)$$

以上 4 个指标的值越大, 表明方法性能越好.

4.4 参数对 RCDMeas 方法有效性的影响

本实验目的在于通过调整 RCDMeas 方法模型参数 ξ , κ , 研究如何提升该方法的检测性能. 设定 $\text{MaxIteration} = 50$, $\delta = 0.5$, $W = 0.5$. 在实验的过程中首先设定 $\kappa = 0.2$, 在此基础上调整参数 ξ 从 0.1 逐渐增大到 0.9, 每一次调整的步长设定为 0.1. 其次, 固定 $\xi = 0.3$, 调整参数 κ 从 0.1 逐渐增大到 0.9, 每一次调整的步长设定为 0.1. 在以上两组参数设定的基础上, 分别采用两组不同的数据集, 使用 RCDMeas 方法开展概念漂移检测, 并基于基准方法的检测结果计算 RCDMeas 方法的各个性能评价指标值. 各个性能评价指标的实验结果如表 2 和 3 所示.

由以上结果可见: (1) 当 κ 固定时, ξ 的值越小, RCDMeas 方法的检测准确率越高. 当 $\xi \leq 0.3$ 时, 方法性能差异不明显. 当 $\xi > 0.3$ 时, 逐渐增大 ξ 的值, RCDMeas 方法的性能衰减明显. 这表明, 当 κ 值固定时, ξ 的值越小, P_ξ 的值越接近最优解. (2) 从实验结果可见, Precision 值较小, 这表明在数据集中, 发生概念漂移的样本数据分布较少. (3) 表 2 中, 在同样参数的设置下, Hit rates 的值略大于 Accuracy, 这表明方法针对非概念漂移的检测结果存在误报, 即部分非概念漂移被误报为概念漂移, 使得 Accuracy 值降低. 表 3 中, κ 值越大这种差异越明显. 这是由于 κ 值越大, Hit rates 的值越小, 同时因为在数据集中概念漂移的样本数据分布较少, 使得这种误报对 Accuracy 值的影响更大. (4) 从表 3

表 2 参数 ξ 对 RCDMeas 方法性能的影响 ($\kappa = 0.2$)

	$\xi = 0.1$	$\xi = 0.2$	$\xi = 0.3$	$\xi = 0.4$	$\xi = 0.5$	$\xi = 0.6$	$\xi = 0.7$	$\xi = 0.8$	$\xi = 0.9$
Accuracy (%)	88.27	87.85	86.94	83.14	79.76	74.26	69.44	63.46	58.23
Precision (%)	42.87	42.26	41.88	39.57	37.42	34.92	31.22	27.33	23.38
Hit rates (%)	89.92	88.24	87.45	84.87	81.33	75.43	70.22	65.17	59.83
F1-score (%)	58.05	57.14	56.63	53.97	51.25	47.73	43.22	38.51	33.62

表 3 参数 κ 对 RCDMeas 方法性能的影响 ($\xi = 0.3$)

	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$	$\kappa = 0.5$	$\kappa = 0.6$	$\kappa = 0.7$	$\kappa = 0.8$	$\kappa = 0.9$
Accuracy (%)	86.01	86.16	86.15	85.73	82.38	73.92	66.46	61.15	56.37
Precision (%)	42.07	41.88	41.23	40.92	36.13	35.53	32.67	29.63	26.88
Hit rates (%)	88.11	87.45	86.22	85.55	84.63	80.20	75.14	68.97	60.43
F1-score (%)	56.94	56.63	55.78	55.36	50.64	49.24	45.53	41.45	37.20

可见, 当 $\kappa > 0.4$ 时, 检测性能明显衰减. 表明当 $\kappa > 0.4$ 时, 表现出明显的漏报. 因此, 我们有必要减小 κ 值, 从而提高概念漂移检测的性能. 当 $\kappa \leq 0.4$ 时, 方法性能差异不明显. 当 $\kappa = 0.1, 0.2$ 时, Accuracy 并没有继续升高, 反而表现出微弱下降. 这是由于部分非概念漂移被错误地检测为概念漂移造成的. 这提醒我们, 需要注意当 κ 值较小时, 尽管概念漂移检测的命中率得到提高, 但容易造成概念漂移检测的误报. 对于本文的应用而言, κ 取值 0.3, 0.4 更有利于提高概念漂移检测的性能.

4.5 性能比较

本实验的目的在于比较不同的可靠性概念漂移检测方法, 以进一步分析 RCDMeas 方法的有效性. 基于以上实验结果, 在本实验中, RCDMeas 方法的参数设定为: MaxIteration = 50, $\delta = 0.5$, $\xi = 0.1$, $\kappa = 0.4$. 在此基础上调整参数 W 从 0.1 逐渐增大到 0.9, 每一次调整的步长设定为 0.1. 本文分别采用两个不同的数据集, 使用 RCDMeas 方法开展概念漂移检测, 并基于基准方法的检测结果计算 RCDMeas 方法的各个性能评价指标值, 在此基础上与 4.2 小节中的 DRD, KLD, MHT 方法进行比较. 不同方法的各个性能评价指标的结果如图 2 所示.

由结果可见, RCDMeas 方法的检测有效性明显高于其他方法, 其 Accuracy 和 Hit rates 值较其他方法约高出 6%~7%, 这表明 Sinkhorn 算法较其他比较的方法更加适用于组合云系统的概念漂移检测问题. 对于 RCDMeas 方法而言, 参数 W 的值对检测结果影响显著, 总体而言, W 的值越大, 概念漂移检测的命中率越低. 当 $W < 0.5$ 时, 性能变化不大. 当 $W > 0.5$ 时, 逐渐增大 W 的值, Hit rates 衰减最为显著, 这表明概念漂移检测的漏报率逐渐升高. 特别地, 当 $W = 0.1, 0.2, 0.3$ 时, 随着 W 的增大, Accuracy 值反而表现出升高的趋势, 这表现出一定的误报率. 当 $W = 0.4$ 时, RCDMeas 方法表现出最优的综合性能.

值得注意的是, 本文通过引入 Sinkhorn 算法, 检测两组可靠性数据流分布的差异, 进而判断组合云系统是否发生可靠性概念漂移. 针对实验过程中比较的 3 种其他方法而言, DRD 和 MHT 方法分别采用均方根误差和平均绝对误差计算数据流分布的距离. KLD 方法则是借助 KL 散度来度量数据流分布的差异. 这 3 类方法尽管计算过程简便易用, 然而当在数据流中相邻时间点数据分布的差异较

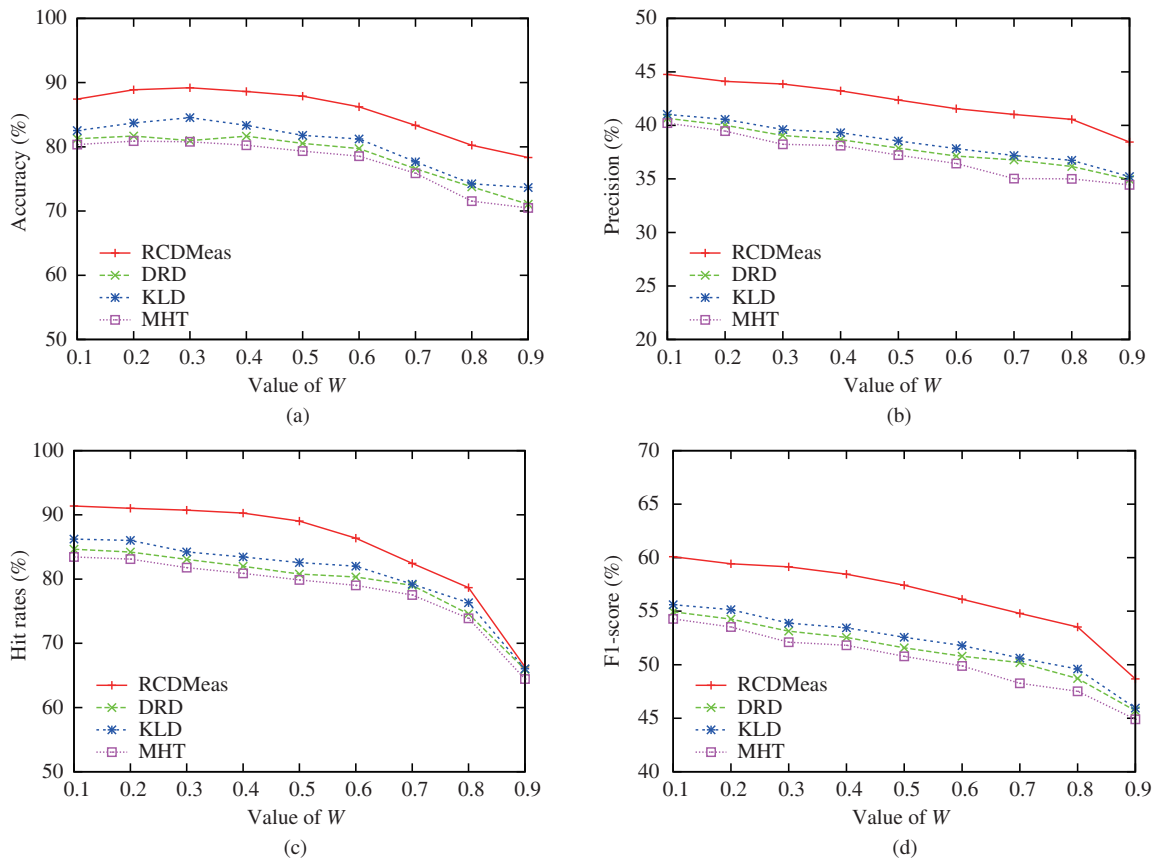


图 2 (网络版彩图) 不同方法的性能比较

Figure 2 (Color online) Performance comparison for different approaches. (a) Accuracy; (b) precision; (c) hit rates; (d) F1-score

小时, 这 3 类计算方法存在不敏感、不精确等方面的不足. 均方根误差和平均绝对误差的计算方法应用的一个前提是需要保证相邻时间点的数据分布之间无相关性. 此外, KL 散度存在非对称性问题, 即 $KL(h||o) \neq KL(o||h)$, 且当两组分布存在重叠 (或差异较小) 时, KLD 方法同样会表现出不敏感性. Sinkhorn 算法建立在严格的统计学理论基础之上, 且在应用过程中, 需要对问题做严格的定义, 其易用性不如以上 3 类方法. 然而这一方法可以很好地应对其他方法在数据流差异度量过程中的不敏感性问题, 具有良好的应用前景.

5 结论

本文针对组合云系统的可靠性概念漂移度量问题, 提出一种系统化的可靠性概念漂移在线度量方法 RCDMeas. 该方法基于 Sinkhorn 距离将可靠性数据流的分布差异检测问题转换为最优运输距离问题, 进而采用熵正则化和不动点迭代求解这一问题. 数据实验结果验证了方法的有效性. 在今后的研究中, 除了继续寻求更为有效的组合云系统概念漂移度量方法, 我们将进一步关注组合云系统的概念漂移自适应方法. 如设计基于环境多样性的组合云系统容错机制, 以应对组合云系统的可靠性概念漂移, 使得组合云系统能够持续、健康地运行.

参考文献

- 1 Huhns M N, Singh M P. Service-oriented computing: key concepts and principles. *IEEE Int Comput*, 2005, 9: 75–81
- 2 Papazoglou M P, van den Heuvel W J. Service oriented architectures: approaches, technologies and research issues. *VLDB J*, 2007, 16: 389–415
- 3 Bouguettaya A, Singh M, Huhns M, et al. A service computing manifesto: the next 10 years. *Commun ACM*, 2017, 60: 64–72
- 4 Ye Z, Bouguettaya A, Zhou X. Economic model-driven cloud service composition. *ACM Trans Int Technol*, 2014, 14: 1–19
- 5 Blake M B, Huhns M N. Web-scale workflow: integrating distributed services. *IEEE Int Comput*, 2008, 12: 55–59
- 6 Duan W X, Hu M, Zhou Q, et al. Reliability in cloud computing system: a review. *J Comput Res Dev*, 2020, 57: 102–123 [段文雪, 胡铭, 周琼, 等. 云计算系统可靠性研究综述. *计算机研究与发展*, 2020, 57: 102–123]
- 7 Jamshidi M. *System of Systems Engineering: Innovations for the Twenty-First Century*. Hoboken: John Wiley & Sons, 2011
- 8 Peng K L, Huang C Y. Reliability analysis of on-demand service-based software systems considering failure dependencies. *IEEE Trans Serv Comput*, 2017, 10: 423–435
- 9 Xue X, Wang S F, Zhang L J, et al. Evaluating of dynamic service matching strategy for social manufacturing in cloud environment. *Future Generation Comput Syst*, 2019, 91: 311–326
- 10 Lu J, Liu A J, Dong F, et al. Learning under concept drift: a review. *IEEE Trans Knowl Data Eng*, 2019, 31: 2346–2363
- 11 Zhao P, Zhou Z H. Distribution-free one-pass learning. 2017. arXiv:1706.02471v
- 12 Du H Y, Wang W J, Bai L. A novel evolving data stream clustering method based on optimization model. *Sci Sin Inform*, 2017, 47: 1464–1482 [杜航原, 王文剑, 白亮. 一种基于优化模型的演化数据流聚类方法. *中国科学: 信息科学*, 2017, 47: 1464–1482]
- 13 Zheng Z B, Lyu M R T, Wang H M. Service fault tolerance for highly reliable service-oriented systems: an overview. *Sci China Inf Sci*, 2015, 58: 052101
- 14 Zheng Z B, Lyu M R T. Collaborative reliability prediction of service-oriented systems. In: *Proceedings of International Conference on Software Engineering (ICSE)*, Cape Town, 2010. 35–44
- 15 Wang H B, Wang L, Yu Q, et al. Online reliability prediction via motifs-based dynamic Bayesian networks for service-oriented systems. *IEEE Trans Softw Eng*, 2017, 43: 556–579
- 16 Salfner F, Lenk M, Malek M. A survey of online failure prediction methods. *ACM Comput Surveys*, 2010, 42: 10
- 17 ANSI/IEEE. IEEE standard glossary of software engineering terminology. ANSI/IEEE STD-610.12-1990, 1990. <https://ieeexplore.ieee.org/document/159342>
- 18 Yamada S, Ohba M, Osaki S. S-shaped reliability growth modeling for software error detection. *IEEE Trans Reliab*, 1983, 32: 475–484
- 19 Brosch F, Koziolk H, Buhnova B, et al. Architecture-based reliability prediction with the palladio component model. *IEEE Trans Softw Eng*, 2011, 38: 1319–1339
- 20 Wang L. Architecture-based reliability-sensitive criticality measure for fault-tolerance cloud applications. *IEEE Trans Parallel Distrib Syst*, 2019, 30: 2408–2421
- 21 Cuturi M. Sinkhorn distances: lightspeed computation of optimal transport. In: *Proceedings of Neural Information Processing Systems*, Lake Tahoe, 2013. 2292–2300
- 22 Peyré G, Cuturi M. Computational optimal transport. *Found Trends Mach Learn*, 2019, 11: 355–206
- 23 Al-Masri E, Mahmoud Q H. Investigating web services on the world wide web. In: *Proceedings of the 17th International Conference on World Wide Web (WWW)*, Beijing, 2008. 795–804
- 24 Jiang W, Lee D, Hu S L. Large-scale longitudinal analysis of soap-based and restful web services. In: *Proceedings of the 19th International Conference on Web Services (ICWS)*, Honolulu, 2012. 218–225

Reliability concept drift online measurement for composite cloud systems

Lei WANG^{1*}, Yunqiu ZHANG¹, Bingfeng XU² & Yiqing XU²

1. Department of Management Science and Engineering, Nanjing Forestry University, Nanjing 210037, China;

2. Department of Software Engineering, Nanjing Forestry University, Nanjing 210037, China

* Corresponding author. E-mail: leiwang@njfu.edu.cn

Abstract Benefiting from the pay-as-you-go business paradigm, the scale of the global cloud computing industry grows rapidly. It has become a possible way to dynamically integrate and cooperate RESTful and SOAP-based component cloud services to develop large-scale complex software systems based on service composition techniques. The constructed composite cloud systems run under dynamic and uncertain operating environments. The issue of how to deal with the concept drift of the reliability data streams to guarantee a stable execution of composite cloud systems has become a grand challenge. To provide some early guidance and act as a trigger mechanism for composite cloud systems' reliability adaptation during the system operation and maintenance, we propose a reliability concept drift measurement approach via Sinkhorn distance for composite cloud systems (or RCDMeas) in this paper. This method employs entropy regularization and fixed-point iteration to identify the reliability concept drift between the historical accumulated reliability flow data and the up-to-date reliability flow data. Large-scale data experiments verified the effectiveness of the proposed approach.

Keywords composite cloud systems, reliability, concept drift, online, execution quality assurance



Lei WANG received his Ph.D. degree in computer science from Southeast University, China. He is an associate professor at Nanjing Forestry University, China. He visited Department of Computer Science and Engineering, The Chinese University of Hong Kong, during 2019. His research interests mainly include service computing, software reliability engineering, and data mining.



Yunqiu ZHANG is currently pursuing her master's degree at Nanjing Forestry University, China. Her research interests mainly include cloud service API recommendation, concept drift detection and optimal transport distances.



Bingfeng XU received her Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, China. She is a lecture at Nanjing Forestry University, China. Her research interests mainly include software engineering and software reliability engineering.



Yiqing XU received his Ph.D. degree in computer science from Southeast University, China. He is a lecture at Nanjing Forestry University, China. His research interests mainly include artificial intelligence and bioinformatics.