

有限干扰下一种稳定的分布式监控算法

严欣愉, 黄增峰

引用本文

严欣愉, 黄增峰. 有限干扰下一种稳定的分布式监控算法[J]. 计算机科学, 2024, 51(11A): 240200050-7.

YAN Xinyu, HUANG Zengfeng. Robust Distributed Monitoring Algorithm Under Limited Interference[J]. Computer Science, 2024, 51(11A): 240200050-7.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[面向云存储的机载软件持有性证明](#)

Airborne Software Provable Data Possession for Cloud Storage

计算机科学, 2024, 51(11A): 240400040-10. <https://doi.org/10.11896/jsjcx.240400040>

[基于关键点密度优化的ORB算法](#)

ORB Algorithm Based on Key Point Density Optimization

计算机科学, 2024, 51(11A): 240300048-5. <https://doi.org/10.11896/jsjcx.240300048>

[面向公平性联邦学习的指纹识别算法](#)

Study on Fingerprint Recognition Algorithm for Fairness in Federated Learning

计算机科学, 2024, 51(6A): 230800043-9. <https://doi.org/10.11896/jsjcx.230800043>

[基于聚簇模型重用的概念漂移数据流半监督分类算法](#)

Semi-supervised Classification of Data Stream with Concept Drift Based on Clustering Model Reuse

计算机科学, 2024, 51(4): 124-131. <https://doi.org/10.11896/jsjcx.230300023>

[基于启发式粗化算法的半监督图神经网络的训练加速框架及算法](#)

Framework and Algorithms for Accelerating Training of Semi-supervised Graph Neural Network Based on Heuristic Coarsening Algorithms

计算机科学, 2024, 51(3): 48-55. <https://doi.org/10.11896/jsjcx.221200158>

有限干扰下一种稳定的分布式监控算法

严欣愉 黄增峰

复旦大学大数据学院 上海 200433

(21210980121@m.fudan.edu.cn)

摘要 分布式监控问题作为分布式系统中的一个热点领域,主要关注如何高效地协调多个传感器与中枢处理器之间的计算任务。与将所有传感器信号实时传输至中枢的低效方法相比,分布式算法通过逐轮统计、汇总结果后再发送给中枢,显然更经济。良好的分布式监控算法能以较小的通信代价完成对特定目标的监控任务,有效节约传感器电能,延长使用寿命。该类算法对通信效率和准确性有双重需求。然而,目前对于基于预设概率分布的分布式阈值监控问题的研究相对有限,且现有研究往往基于理想化的假设,导致所设计的算法对实际干扰缺乏抵抗能力。通过引入干扰因素来模拟现实世界的复杂性,旨在寻找更为鲁棒的分布式监控算法。所提算法通过合理选择通信时机,不仅减少了通信次数,显著降低了通信代价,同时也保证了有干扰的环境下的分布式算法的准确性。所提算法的准确性可以在理论层面得到证明,其通信代价在干扰较少时可达到 $O(K \log \log N)$ 。这一研究为分布式监控算法提供了新的视角,为现实复杂监控问题的解决提供了有力支持。

关键词: 随机算法; 抽样; 分布式监控; 数据流; 数值监控

中图分类号 TP311

Robust Distributed Monitoring Algorithm Under Limited Interference

YAN Xinyu and HUANG Zengfeng

School of Data Science, Fudan University, Shanghai 200433, China

Abstract Distributed monitoring is a pivotal area in the field of distributed systems. It focuses on the coordination of computational tasks between multiple sensors and a central processor. Typically, sensors notify the central processor immediately upon receiving signals, leading to energy wastage. However, this traditional communication mechanism is inefficient for several reasons. Firstly, the central processor often only needs summarized information, like the total number of signals received over a period. Secondly, sensors buried within objects rely on battery power, making replacements challenging. Lastly, the energy consumed in communication surpasses that needed for computation. In contrast, distributed algorithms summarize results before transmitting to the central processor, proving to be more economical. Good distributed tracking algorithms not only achieve tracking tasks with smaller communication costs, conserving sensor energy and prolonging lifespan, but also demand considerations for communication efficiency and accuracy. However, current research on distributed threshold monitoring problems based on preset probability distributions is relatively limited. Existing studies often rely on idealized assumptions, resulting in algorithms lacking robustness against real-world interference. This paper introduces interference to simulate the complexities of the real world, aiming to identify more robust distributed tracking algorithms. The proposed algorithm reduces communication rounds by judiciously selecting the thresholds for sensors to send notifications to the central processor, significantly reducing communication costs. Additionally, it ensures algorithm accuracy in the presence of interference. The algorithm's accuracy is theoretically proven, while its communication cost can reach $O(K \log \log N)$ when interference is limited. This study provides a fresh perspective on distributed tracking algorithms, supporting the solution of practical tracking problems.

Keywords Random algorithm, Sampling, Distributed monitoring, Data streams, Functional monitoring

1 引言

在分布式系统的研究领域中,分布式监控问题占据了重要地位。设想一个由众多分布式传感器和一个中央处理器构成的系统,这些传感器散布在不同的地理位置,负责将采集到的数据传输给中央处理器以供进一步的分析和处理。在传统的通信模式中,传感器一旦探测到信号便立即向中央处理器报告,但这种做法往往会导致不必要的能量消耗^[1]。首先,中央处理器实际上并不需要传感器提供的详尽数据,而是更倾

向于接收概括性的计算结果,例如在一定时间窗口内信号的累积计数。其次,对于那些嵌入于物体内部的传感器,它们通常依赖有限的电池能量,且难以轻松更换^[2]。此外,通信过程中的能量消耗通常远超过数据处理所需的能量。因此,一种高效的分布式监控算法能够通过最小化通信次数或比特数,将所需的结果迅速传送至中枢,从而使中枢能够及时准确地完成计算任务。在这一背景下,本文旨在探讨并提出一种分布式监控算法,以满足系统中对于通信效率和计算准确性的双重需求。

对于分布式监控问题,存在多种场景假设,其中一些常用假设与特征为:

1) 顺序性^[3]:考虑一系列事件 $E = (E_1, E_2, \dots, E_n)$, 这些事件按照一定的顺序发生,并且每个事件只能由传感器 $S = (S_1, S_2, \dots, S_n)$ 中的一个观测到并记录。

2) 间隔足够^[4]:在事件发生的间隔中,传感器与中枢有足够的时间进行通信、计算和调整算法,确保监控系统的有效性。

3) 阈值监控^[3]:中枢维护一个监控函数 $f: 2^E \rightarrow \mathbb{R}$, 该函数用于汇总事件的发生情况。中枢在监控函数达到预设阈值时发出警报提示。通常,监控函数设计简单,例如统计事件总数、监控各种事件的数量或比重^[5-7]、监控发生次数最多的事件^[8],或者统计事件流中不同事件的总类数^[9]。

4) 最小化通信代价:分布式监控算法的目标是最小化通信代价,通信代价通常以整个算法过程的通信次数或者通信总比特数进行统计。

本文的研究基于以下场景,即在大部分时间内,任何事件 E_i 被某个传感器 S_k 观测到的概率是一个确定的值 μ_k ,而在少数时间内存在扰动,此时事件被某个传感器观测到的概率变得不确定。可以参考以下现实中的例子:

1) 在大型设备如飞机中埋设传感器,用于监控异常情况。异常次数累积到一定阈值时进行检修。通常情况下,特定部位出现异常的概率是确定的。

2) 监控网络中多个相关联的站点的访问情况,统计总访问量^[10-11]。当总访问量达到阈值时,可能需要进行扩容等操作。通常情况下,特定站点每个时刻产生的访问量是一个确定的值。

3) 监控交通情况,记录每个路口的车流量。当总车流量达到阈值时,需要进行交通疏导。通常情况下,每个路口每个时刻的车流量是一个确定的值。

本文的贡献在于提出了一种允许一定程度干扰的分布式监控算法。该算法在扰动较小的情况下的效率与无扰动时相当。在扰动较小的假设下,我们提供了算法效率与准确性的详细证明,为解决实际应用中的复杂监控问题提供了有力支持。

2 相关工作

2.1 分布式监控算法的相关工作

在分布式系统中,由于资源受限,将端点接收的全部资源传送至中枢既难以实现又浪费时间。通常,分布式系统注重有效分配任务和负载均衡,为此涌现出了大量的相关算法^[12-13]。相比之下,分布式监控算法的目标更为简单,仅关注以高效方式汇总数据流的重要特征^[14],例如统计多个站点的传感器读数、网络使用日志或网络数据包的情况等^[15-16]。

在此背景下,分布式监控问题吸引了广泛的研究兴趣,并产生了多种算法。Babcock 等提出的 top- k 监控问题^[8],旨在实时追踪动态数值中的前 k 个最大值,其算法仅在触发约束条件时进行通信,有效降低了通信成本。此外,针对连续数据流,研究者提出了包括集合运算和排序估计在内的复杂监控任务^[5,17],Ganguly 等的工作通过哈希草图技术实现了高效、低误差的数据流集合估计^[5]。

Cormode 等进一步研究了分布式数据流上的分布监控问题,通过追踪数值的近似分位数来描述数据流的分布^[6,18]。他们的算法结合了本地追踪和预测模型,实现了通信和存储

的高效性,同时保证了分位数估计的准确性和实时性^[6]。Cormode 还研究了分布式数据流中事件数量和的监控问题,提出了著名的 CMY 算法^[4],并为分布式监控问题的通信代价提供了理论界限。

在 Cormode 等的工作基础上,Wu 等引入了基于概率假设的分布式监控问题,通过预设事件的概率分布来减少通信成本,并在概率分布未知的情况下,通过 CMY 算法对概率分布进行估计,确保了算法的准确性和效率^[19]。

在 Wu 等的研究基础之上,本文提出了一种鲁棒性更好的分布式监控算法,该算法在数据流遭遇干扰并偏离预设概率分布时,仍能维持高效的性能。所提算法通过在每一轮统计过程中引入冗余项,增强了算法对干扰的容忍度,从而显著提升了算法的整体稳定性。通过理论推导,我们证实了算法在面对不确定干扰时的准确性和高效性。这一创新不仅丰富了分布式监控算法的研究领域,也为实际应用中的系统稳定性和效率提供了有力的保障。

2.2 分布式阈值监控问题描述与经典算法

考虑一个由 K 个玩家和一个协调员组成的系统。在这个系统中,每个时刻会有一张牌被随机分发给其中一个玩家。协调员并不知道牌何时发出,也不知道在特定时刻哪个玩家收到了牌。然而,协调员需要在所有玩家收到的牌总和达到阈值 N 时准确发出提示,也就是确定第 N 张牌发出的确切时刻。值得注意的是,每个玩家只能与协调员进行双向通信。那么,为了实现这一目标,协调员与玩家之间最少需要多少次通信?

目前已有大量研究围绕这个问题及其变种展开,例如允许协调员在第 N 张牌发出前给出提示,但必须在 $(1-\epsilon)N$ 张牌发出后^[4],或是让协调员在任意时刻估算已经发出的牌总数^[3,20],其中以 CMY 算法最为经典。

2.3 CMY 算法介绍

针对这一经典的分布式阈值监控问题,Cormode 等提出了一种创新的算法,后续被称为 CMY 算法,以 3 位作者的首字母命名^[4]。CMY 算法在解决分布式监控问题方面取得了重大突破,它通过精巧的设计显著提升了通信效率,为最小化监控过程中所需的通信次数提供了新的视角和方法。

算法 1 CMY 算法^[4]

输入:玩家数 K , 阈值 N 和一个每个时刻向随机玩家发牌的黑盒

输出:在第 N 张牌被玩家接收后,协调员发出提示

1. 如果 $N \leq K$, 每个玩家收到牌后立即报告协调员。协调员在收到 N 次报告后给出提示。
2. 每个玩家维护一个计数器 $c_i, i=1, 2, \dots, K$, 用于记录还未向协调员报告的牌数。初始时,所有计数器为 0。
3. 当一个玩家收到一张牌后,将自身计数器加 1, $c_i \leftarrow c_i + 1$ 。
4. 如果此时某个玩家 i 的计数器达到了 $\lfloor \frac{N}{2K} \rfloor$, 则通知协调员。
5. 当协调员在这一轮中收到了 K 次通知后,统计所有玩家的计数器之和, $c = c_1 + c_2 + \dots + c_K$, 然后更新阈值 $N \leftarrow N - c$ 。协调员将新的阈值 N 发送给每个玩家,玩家的计数器 c_i 全部清零,回到第 1 步开启下一轮统计。

CMY 算法每一轮中,只有第 4, 5 步需要进行一共 $3K$ 次通信。每一轮至少使阈值减少 $\frac{N}{2K} \cdot K = \frac{N}{2}$, 即后一轮的阈值不超过前一轮的 $\frac{1}{2}$ 倍,一共需要 $\log_2 \frac{N}{K}$ 轮将剩余牌数降低到 K 以下,因此通信代价为 $O\left(K \log \frac{N}{K}\right)$ 。Cormode 同时也证

明了这一通信代价是此类分布式通信问题的下界,即最坏情况下,通信代价不会低于 $O\left(K \log \frac{N}{K}\right)$ 。

2.4 有概率假设的分布式监控问题

Wu 等提出了有概率假设的分布式监控问题^[19],即每个时刻玩家 i 能获得牌的概率为 $\mu_i, i=1,2,\dots,K$ 且 $\sum_{i=1}^K \mu_i = 1$ 。其中,每个玩家的概率 μ_i 是确定且已知的,不随时刻发生变化。

算法 2 基于预设概率的分布式监控算法^[19]

输入:玩家数 K , 阈值 N 和一个每个时刻向随机玩家发牌的黑盒

输出:在第 N 张牌被玩家接收后,协调员发出提示

1. 如果 $N \leq \beta K \ln \frac{K}{\delta}$, 执行 CMY 算法; 否则, 继续执行以下步骤。
2. 每个玩家维护一个计数器 $c_i, i=1,2,\dots,K$, 用于统计一轮中获得的牌数。初始时, 所有计数器为 0。
3. 协调员向每个玩家 i 发送一个该轮阈值 s_i 。
4. 当一个玩家收到一张牌后, 将自身计数器加 1, $c_i \leftarrow c_i + 1$ 。
5. 如果此时某个玩家 i 的计数器满足了某个与阈值 s_i 有关的条件, 通知协调员。协调员在收到了 1 次通知后结束这一轮统计, 并统计所有玩家这一轮获得的牌数之和, 即 $c = \sum_{i=1}^K c_i$ 。更新 $N, N \leftarrow N - c$ 。
6. 所有玩家将自己的计数器 c_i 置 0, 回到第 1 步, 开始下一轮统计。

Wu 通过数学证明, 在每一轮监控中, 通过为每个玩家设定恰当的阈值 s_i , 当任一玩家的计数器值达到其阈值 s_i 时, 剩余的牌数阈值都会从 N 下降至 $O(\sqrt{N})$ 的量级, 最终证明了算法 2 能够在通信代价接近 $O(K \log \log N)$ 的情况下, 有效解决基于概率假设的分布式监控问题。

尽管算法 2 在理想条件下表现出色, 然而在实际应用中, 预设的概率分布可能因外部干扰和内部变化而发生波动, 并不总能保持稳定。特别是当这些干扰集中发生在数据流的尾部时, 它们可能对算法 2 的通信效率产生显著的负面影响, 导致通信代价大幅增加, 下一章的实验中将介绍这一点。为了解决这一问题, 本文将在后续章节中详细介绍一种改进后的算法, 该算法在算法 2 的基础上增强了对干扰的鲁棒性, 能够在面对概率波动时保持高效的通信性能。

3 有干扰的分布式监控问题与算法

3.1 问题描述

考虑一个由 K 个玩家和一个协调员组成的系统, 在这个系统中, 每个时刻都有一张牌会被分发给其中一个玩家。对于大多数时刻, 玩家 i 能获得牌的概率为 $\mu_i, i=1,2,\dots,K$ 且 $\sum_{i=1}^K \mu_i = 1, \mu_i$ 是确定且已知的。在另一小部分时刻, 由于干扰的影响, 概率变得不确定。协调员既不知道牌何时发出, 也不知道在特定时刻哪个玩家收到了牌, 更不知道哪些时刻的概率受到了干扰。然而, 协调员需要在所有玩家收到的牌总和达到阈值 N 时准确发出提示, 也就是确定第 N 张牌被接收的确切时刻。每个玩家只能与协调员进行双向通信。那么, 为了实现这一目标, 协调员与玩家间最少需要多少次通信?

3.2 干扰对基于预设概率的分布式监控算法稳定性的影响

考虑这样一个包含干扰的分布式监控问题。牌数阈值 N 为 10^7 , 玩家人数 $K=32$ 。在不受干扰的时刻, 所有玩家每个时刻收到牌的概率相同, 都为 $\mu_i = 1/32, i=1,2,\dots,32$ 。在 $\Delta = \sqrt{N}$ 个受干扰的时刻中, 概率分布发生变化, 只有 1 号玩家能收到牌。当这些干扰时刻均匀分布在数据流的尾部时,

算法 2 的表现如图 1 所示。

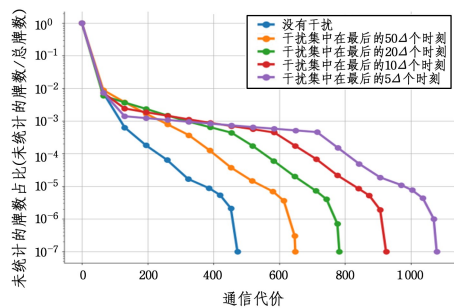


图 1 算法 2 在不同干扰分布下的运行轨迹

Fig. 1 Algorithm 2's trajectory under different interference distributions

图 1 横轴代表到目前为止发生的通信次数, 纵轴则表示剩余的牌数比重。由图可知, 随着干扰时刻集中在数据流的尾部, 算法 2 所需的通信代价急剧上升。尽管干扰次数相对于总牌数 N 的比例很小 (仅为 $1/\sqrt{N}$), 但其对算法的通信效率产生了重大影响。这一发现突显了算法 2 在抗干扰能力上的不足, 强调了开发具有更高稳定性的分布式监控算法的重要性。

3.3 考虑干扰的分布式监控算法设计

尽管算法 2 在应对受干扰的分布式监控问题时遭遇了极大的困难, 但它在利用预设的概率分布来优化性能方面仍展现出了重要优势。问题主要出在少量但集中的干扰破坏了某一段时间内的概率分布, 大大影响了算法 2 的性能。基于此, 我们可以提出一个合理的算法设计方向, 对算法 2 进行两项关键改进: 1) 每个统计轮次的阈值 s_i 应根据干扰的频率进行调整, 以保证在任何一轮中, 干扰事件的数量都不会对预设的概率分布产生重大影响; 2) 进入 CMY 阶段的时机需要重新设定, 确保即使在阈值 N 降至较低水平且干扰密度较高的情况下, 通信效率至少仍能保持 CMY 算法的水平。

算法 3 带有 Δ 次干扰的分布式监控算法

输入: 玩家数 K , 阈值 N 以及一个黑盒, 该黑盒在大多数时刻以概率 $\mu_i, i=1,2,\dots,K$ 发牌给玩家 i , 而在 Δ 个干扰时刻, 则向随机玩家发牌

输出: 在第 N 张牌被玩家接收后, 协调员发出提示

1. 如果 $N \leq \max\left\{\frac{\Delta^2}{\ln\left(\frac{K}{\delta}\right)}, 2\left(\sqrt{2} + \frac{5}{3}\right)^2 K \ln \frac{K}{\delta}\right\}$, 执行 CMY 算法。
2. 每个玩家维护一个计数器 $c_i, i=1,2,\dots,K$, 用于统计每一轮获得的牌数。初始时, 所有计数器为 0。
3. 协调员向每个玩家 i 发送一个该轮阈值 s_i 。
4. 当一个玩家收到一张牌后, 将自身计数器加 1, $c_i \leftarrow c_i + 1$ 。
5. 如果此时某个玩家 i 的计数器达到了 s_i , 通知协调员。协调员立刻结束这一轮的统计, 并统计所有玩家这一轮获得的牌数之和, 即 $c = \sum_{i=1}^K c_i$ 。更新 $N, N \leftarrow N - c$ 。
6. 所有玩家将自己的计数器 c_i 置 0, 回到第 1 步, 开始下一轮统计。

算法中 s_i 取值如下:

$$s_i = \frac{2}{3} \ln \frac{K}{\delta} + \sqrt{2\mu_i(1-\mu_i)t \ln \frac{K}{\delta} + t - (1-\mu_i)(t-\Delta)} \quad (1)$$

其中

$$t = N - \left(\sqrt{2} + \frac{5}{3}\right) \sqrt{KN \ln \frac{K}{\delta}} \quad (2)$$

定理 1 算法 3 保证了协调员以不低于 $1 - C\delta$ 的概率在

$$O(K \log \log N + K \log \left(\max \left\{ \frac{\Delta^2}{\ln \left(\frac{k}{\delta} \right)}, 2 \left(\sqrt{2} + \frac{5}{3} \right)^2 K \ln \frac{K}{\delta} \right\} \right))$$

次通信中,找到第 N 张牌被接收的时刻。当干扰次数较少时,通信代价接近 $O(K \log \log N)$ 。而当干扰次数较多时,通信代价接近 $O(K \log \Delta)$ 。后文将证明这一结论。

3.4 算法的通信代价与准确性证明

3.4.1 玩家获得牌数的上下界

引理 1 如果每个玩家 i 在任何时刻都有概率 μ_i 获得一张牌,那么在总共发出 t 张牌时,每个玩家获得的牌数 X_i 具有以下上下界:

$$UB_i(t) = \mu_i t + \frac{1}{3} \ln \frac{K}{\delta} + \sqrt{\frac{1}{9} \ln^2 \frac{K}{\delta} + 2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} \quad (3)$$

$$LB_i(t) = \mu_i t - \frac{1}{3} \ln \frac{K}{\delta} - \sqrt{\frac{1}{9} \ln^2 \frac{K}{\delta} + 2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} \quad (4)$$

X_i 以不超过 $\frac{\delta}{K}$ 的概率大于 UB_i , 同样地, X_i 以不超过 $\frac{\delta}{K}$ 的概率小于 LB_i 。

引理 1 证明: 假设随机变量 $X_{i,s}, i=1, 2, \dots, K, s=1, 2, \dots, t$ 表示玩家 i 在时刻 s 是否收到一张牌。 $X_{i,s}$ 服从伯努利分布, 其中 $P(X_{i,s}=1) = \mu_i$ 且 $P(X_{i,s}=0) = 1 - \mu_i$ 。定义 $X_i = \sum_{s=1}^t X_{i,s}$ 。应用 Bernstein 不等式^[21], 可以得到 X_i 的上界。

$$P[X_i \geq E[X_i] + \lambda] \leq \exp \left(- \frac{\lambda^2}{2 \left(\text{Var}[X_i] + \frac{M\lambda}{3} \right)} \right)$$

其中, X_i 的期望 $E[X_i] = \mu_i t$, 方差 $\text{Var}[X_i] = \mu_i(1-\mu_i)t$, M 是一个满足 $X_i \leq E[X_i] + M, \forall i \in [t]$ 的正常数。此处取 $M=1$ 即可。为了使不等式右侧等于 $\frac{\delta}{K}$, 需让

$$\exp \left(- \frac{\lambda^2}{2 \left(\text{Var}[X_i] + \frac{M\lambda}{3} \right)} \right) = \frac{\delta}{K}, \text{ 展开整理后得到关于 } \lambda \text{ 的二次方程:}$$

$$\lambda^2 - \frac{2}{3} \ln \frac{K}{\delta} \lambda - 2\mu_i(1-\mu_i)t \ln \frac{K}{\delta} = 0$$

解得:

$$\lambda = \frac{1}{3} \ln \frac{K}{\delta} + \sqrt{\frac{1}{9} \ln^2 \frac{K}{\delta} + 2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}}$$

引理 1 中的上界部分 $P(X_i \geq UB_i(t)) \leq \frac{\delta}{K}$ 得证。同理

可证 $P(X_i \leq LB_i(t)) \leq \frac{\delta}{K}$ 。

3.4.2 算法 3 总轮次数

考虑一轮统计中前 t 个发牌时刻, 假设随机变量 $X_{i,s}, i=1, 2, \dots, K, s=1, 2, \dots, t$ 表示玩家 i 在时刻 s 是否收到一张牌。则此时玩家获得的牌数 $c_i = \sum_{s=1}^t X_{i,s} = \sum_{s \in I} X_{i,s} + \sum_{s \in I^c} X_{i,s}$, 其中 I 是没有受到干扰的集合。由引理 1 可知 $P \left(\sum_{s \in I} X_{i,s} \geq UB_i(|I|) \right) \leq \frac{\delta}{K}$, 又因为 $\sum_{s \in I} X_{i,s} \leq t - |I|$, 所以有不低于 $1 - \frac{\delta}{K}$ 的概率, $c_i \leq UB_i(|I|) + (t - |I|)$, 代入式(3)后可得 $c_i \leq$

$$\mu_i |I| + \frac{1}{3} \ln \frac{K}{\delta} + \sqrt{\frac{1}{9} \ln^2 \frac{K}{\delta} + 2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} + (t - |I|)。$$

考虑到 t 个时刻内至少有 Δ 个受到干扰, 即 $|I| \in [t - \Delta, t]$, 化简后得:

$$c_i \leq \frac{2}{3} \ln \frac{K}{\delta} + \sqrt{2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} + t - (1-\mu_i)(t - \Delta) \quad (5)$$

不等式(5)的右侧等于式(1)中的 s_i 。因此, 有不低于 $1 - \frac{\delta}{K}$ 的概率 $c_i \leq s_i$ 。

引理 2 如果在一轮中将玩家的阈值 s_i 和 t 设置为式(1)和式(2)中的值, 那么在这一轮结束后, 以至少 $1 - \delta$ 的概率剩余牌数 N 至少减少到 $\left(\sqrt{2} + \frac{5}{3} \right) \sqrt{kN \ln \frac{K}{\delta}}$, 并且协调员不会错过阈值 N 。

引理 2 证明: 由于在一轮统计中前 t 个发牌时刻, 有不低于 $1 - \frac{\delta}{K}$ 的概率玩家 i 的计数器 $n_i \leq s_i$ 。换句话说, 以至多 $\frac{\delta}{K}$ 的概率, 玩家 i 向协调员发出通知。因此, 在前 t 个发牌时刻中, 所有的 K 个玩家中至少有一个玩家向协调员发出通知的概率至多为 δ 。因此, 以至少 $1 - \delta$ 的概率, 剩余牌数 N 至少减少到 $N - t = \left(\sqrt{2} + \frac{5}{3} \right) \sqrt{kN \ln \frac{K}{\delta}}$ 。另一方面, 一轮中所有玩家能收到的牌数至多为 $\sum_i s_i$ 。代入式(1)中的 s_i , 得:

$$\begin{aligned} \sum_{i=1}^K s_i &= \frac{2}{3} K \ln \frac{K}{\delta} + \sum_{i=1}^K \sqrt{2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} + Kt - (K - \sum_{i=1}^K \mu_i)(t - \Delta) \\ &= t + (K-1)\Delta + \frac{2}{3} K \ln \frac{K}{\delta} + \sum_{i=1}^K \sqrt{2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} \end{aligned}$$

结合平方根函数的凸性, 有:

$$\begin{aligned} \sum_{i=1}^K \frac{1}{K} \sqrt{2\mu_i(1-\mu_i)t \ln \frac{K}{\delta}} &\leq \sqrt{\sum_{i=1}^K \frac{1}{K} \frac{1}{K} \left(2\mu_i(1-\mu_i)t \ln \frac{K}{\delta} \right)} \\ &\leq \sqrt{\sum_{i=1}^K \frac{2\mu_i t \ln \frac{K}{\delta}}{K}} = \sqrt{\frac{2t \ln K / \delta}{K}} \end{aligned}$$

代入 $\sum_{i=1}^K s_i$ 中得到:

$$\sum_{i=1}^K s_i \leq t + (K-1)\Delta + \frac{2}{3} K \ln \frac{K}{\delta} + \sqrt{2Kt \ln \frac{K}{\delta}} \quad (6)$$

又因为 $N > \max \left\{ \frac{\Delta^2}{\ln \left(\frac{k}{\delta} \right)}, 2 \left(\sqrt{2} + \frac{5}{3} \right)^2 K \ln \frac{K}{\delta} \right\}$ (否则

执行 CMY 算法), 所以 $K\Delta \leq \sqrt{kN \ln \frac{K}{\delta}}$ 且 $K \ln \frac{K}{\delta} \leq$

$\sqrt{kN \ln \frac{K}{\delta}}$, 代入式(6)中可得:

$$\begin{aligned} \sum_{i=1}^K s_i &\leq t + \sqrt{kN \ln \frac{K}{\delta}} + \frac{2}{3} \sqrt{kN \ln \frac{K}{\delta}} + \sqrt{2kN \ln \frac{K}{\delta}} \\ &= t + \left(\frac{5}{3} + \sqrt{2} \right) \sqrt{kN \ln \frac{K}{\delta}} \end{aligned}$$

代入式(2)中的 t 可知 $\sum_i s_i \leq N$, 这说明协调员不会在一轮中错过阈值 N 。引理 2 得证。

由引理 2 可知, 算法 3 的每一轮统计, 以至少 $1 - \delta$ 的概率将剩余牌数 N 至少减少到 $\left(\sqrt{2} + \frac{5}{3} \right) \sqrt{kN \ln \frac{K}{\delta}}$ 。列出递归关系:

$$N_0 = N$$

$$N_r = \left(\sqrt{2} + \frac{5}{3}\right) \sqrt{K N_{r-1} \ln \frac{K}{\delta}}$$

通过递归可知:

$$\begin{aligned} N_r &= \left(\sqrt{2} + \frac{5}{3}\right) \sqrt{K \ln \frac{K}{\delta}} N_{r-1}^{\frac{1}{2}} \\ &= \left(\left(\sqrt{2} + \frac{5}{3}\right) \sqrt{K \ln \frac{K}{\delta}}\right)^{1+\frac{1}{2}} \cdot N_{r-2}^{\frac{1}{4}} \cdots \\ &= \left(\left(\sqrt{2} + \frac{5}{3}\right) \sqrt{K \ln \frac{K}{\delta}}\right)^{1+\frac{1}{2}+\cdots+\frac{1}{2^{r-2}}} \cdot N_0^{\frac{1}{2^r}} \end{aligned}$$

当 $r = \log_2 \log_2 N$ 时, $N_r < \left(\left(\sqrt{2} + \frac{5}{3}\right) \sqrt{K \ln \frac{K}{\delta}}\right)^2 \cdot 2 = 2 \left(\sqrt{2} + \frac{5}{3}\right)^2 K \ln \frac{K}{\delta}$, 此时 N 已满足步骤 1 中的条件, 循环结束。因此, 至多经过 $\log_2 \log_2 N$ 轮统计后, 算法 3 开始执行 CMY 算法。

3.4.3 算法 3 的准确性与通信代价

由引理 2 可知, 算法 3 的每一轮统计以至少 $1 - \delta$ 的概率将剩余牌数 N 至少减少到 $\left(\sqrt{2} + \frac{5}{3}\right) \sqrt{k N \ln \frac{K}{\delta}}$, 因此 $\log_2 \log_2 N$ 轮后, 以至少 $(1 - \delta) \cdot \log_2 \log_2 N$ 的概率将剩余牌数 N 减少到步骤 1 中的条件。

算法 3 中的通信代价包含两部分。一是步骤 1 中 CMY 算法所需的通信次数, 总共通信代价为 $O\left(K \log\left(\max\left\{\frac{\Delta^2}{\ln\left(\frac{k}{\delta}\right)}, 2 \left(\sqrt{2} + \frac{5}{3}\right)^2 K \ln \frac{K}{\delta}\right\} / K\right)\right)$ 。二是每一轮步骤 2-6 中的通信代价, 其中步骤 3 需要 K 次通信, 步骤 5 需要 K 次通信。所以在 $\log_2 \log_2 N$ 轮中, 需要的通信代价为 $O(K \log \log N)$ 。因此算法 3 所需的通信代价为

$$O\left(K \log \log N + K \log\left(\max\left\{\frac{\Delta^2}{\ln\left(\frac{k}{\delta}\right)}, 2 \left(\sqrt{2} + \frac{5}{3}\right)^2 K \ln \frac{K}{\delta}\right\} / K\right)\right)$$

当干扰次数满足 $\Delta = O(\log N)$ 时, 通信代价可以简化为 $O(K \log \log N)$, 即算法 3 可以容忍较少的干扰次数。而当干扰次数较多时, 即 $\Delta > O(\log N)$ 时, 通信代价接近 $O(K \log \Delta)$ 。至此, 定理 1 证明完毕。

3.4.4 算法 3 通信代价下界的讨论

如上文所述, 当干扰次数 Δ 较大时, 即 $\Delta > O(\log N)$ 时, 通信代价接近 $O(K \log \Delta)$ 。直观来看, 是因为受到干扰的时刻过多从而导致通信代价中后一项的影响更大。事实上, 通信代价不会低于 $O(K \log \Delta)$ 。设想一种极端情况, 如果所有的受扰动时刻都在最后出现, 即最后一张牌的发放是随机的, 不满足任何分布。Cormode 等已经证明了, 分布式监控张随机发放的牌需要的通信代价下界为 $\Omega\left(K \log \frac{\Delta}{K}\right)$ ^[4]。因此, $O(K \log \Delta)$ 的通信代价已经接近该问题的下界。

4 实验对比

本章将对 3 种不同算法在处理受干扰的分布式监控问题时的性能进行比较分析。这 3 种算法包括: Cormode 提出的算法 1 (CMY 算法)、Wu 提出的算法 2 (基于预设概率分布的分布式监控算法), 以及本文提出的算法 3 (带有 Δ 次干扰的分布式监控算法)。通过实验比较, 我们旨在清晰直观地展示这些算法在面对不同干扰模式时的性能表现。

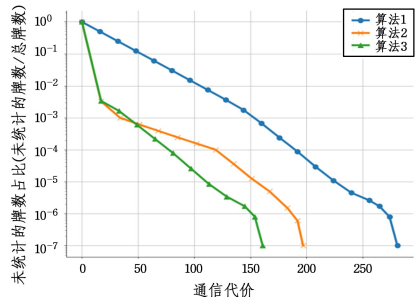
4.1 干扰集中在最后时段对算法性能的影响

4.1.1 实验设置

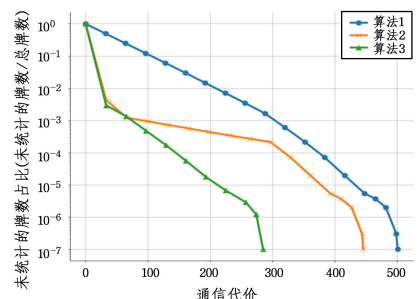
实验设置如下: 玩家人数分别为 8, 16, 32, 牌数阈值 N 设定为 10^7 , 干扰次数 $\Delta = \lfloor \sqrt{N} \rfloor$ 。在不受干扰的时刻, 所有玩家每个时刻收到牌的概率相同, 都为 $\mu_i = 1/K, i = 1, 2, \dots, 32$ 。在 $\Delta = \sqrt{N}$ 个受干扰的时刻中, 只有 1 号玩家能收到牌。干扰均匀分布在最后的 5Δ 个时刻中, 即在最后 5Δ 个发牌时刻中, 每 5 个时刻中就有一个时刻受到干扰。

4.1.2 实验结果与分析

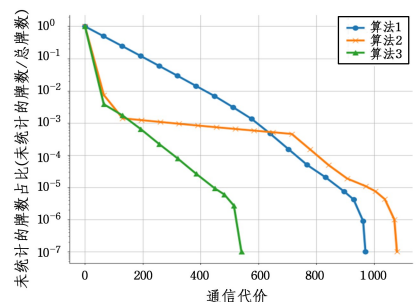
图 2(a)~图 2(c) 分别给出了不同玩家人数时 3 种分布式监控算法的运行情况。横轴表示到目前为止发生的通信次数, 纵轴则代表剩余的牌数比重 (即未发出的牌数与 N 的比值)。从这 3 张图中可以观察到, 当 Δ 个干扰集中在最后 5Δ 个时刻时, 算法 1 (蓝线) 和算法 3 (绿线) 的运行轨迹总体上保持相对稳定。然而, 算法 2 (黄线) 的表现在进入受干扰时段后迅速变差, 其轨迹下降速度显著减缓, 通信代价在短时间内急剧增加。特别是在图 2(c) 中, 算法 2 的总通信代价甚至超过了不需要预设概率的算法 1 (CMY 算法)。这一结果明确显示了算法 3 在稳定性方面明显优于算法 2。



(a) 玩家人数 $K=8$ 时 3 种算法的运行轨迹



(b) 玩家人数 $K=16$ 时 3 种算法的运行轨迹



(c) 玩家人数 $K=32$ 时 3 种算法的运行轨迹

图 2 不同玩家人数时 3 种分布式监控算法的运行轨迹
(电子版为彩图)

Fig. 2 Trajectories of three distributed monitoring algorithms with different number of players

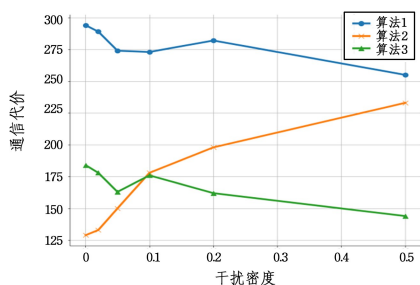
4.2 不同干扰密度对算法性能的影响

4.2.1 实验设置

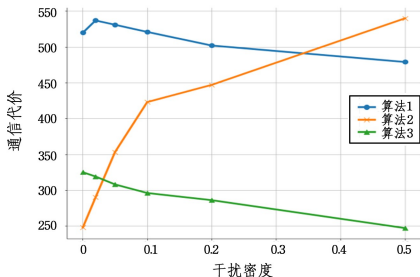
实验设置如下: 玩家人数分别为 8, 16, 32, 牌数阈值 N 设定为 10^7 , 干扰次数 $\Delta = \lfloor \sqrt{N} \rfloor$ 。在不受干扰的时刻, 所有玩家每个时刻收到牌的概率相同, 都为 $\mu_i = 1/K, i = 1, 2, \dots, 32$ 。在 $\Delta = \sqrt{N}$ 个受干扰的时刻中, 只有 1 号玩家能收到牌。每轮实验记录 6 个观测点, 分别对应 Δ 次干扰均匀分布在所有时刻中, 以及均匀分布在最后的 $50\Delta, 20\Delta, 10\Delta, 5\Delta, 2\Delta$ 中。这些观测点对应的干扰密度 (横坐标) 分别为 $1/\sqrt{N} \approx 0, 1/50, 1/20, 1/10, 1/5, 1/2$, 而纵坐标表示在相应干扰密度下算法总共所需的通信代价。通过这样的实验设置, 我们旨在深入分析各种分布式监控算法在不同干扰密度下的通信效率。

4.2.2 实验结果与分析

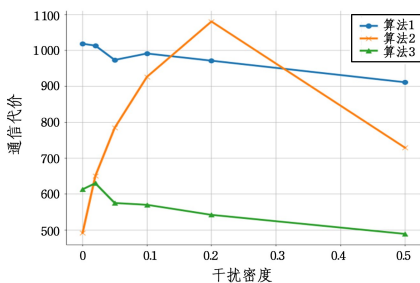
图 3(a)~图 3(c) 直观地呈现了 3 种算法在不同干扰密度条件下的性能变化趋势。



(a) 玩家人数 $K=8$ 时 3 种算法的性能与干扰密度的关系



(b) 玩家人数 $K=16$ 时 3 种算法的性能与干扰密度的关系



(c) 玩家人数 $K=32$ 时 3 种算法的性能与干扰密度的关系

图 3 不同玩家人数时 3 种算法的性能与干扰密度的关系
(电子版为彩图)

Fig. 3 Performance of three algorithms versus interference density with different number of players

算法 1 (蓝线) 与算法 3 (绿线) 都显示出较好的稳定性, 其通信代价几乎不受干扰密度变化的影响。与此相反, 算法 2 (黄线) 的性能对于干扰密度非常敏感, 随着干扰时刻集中在数据流尾部, 其通信代价出现了明显的波动。特别是在干扰密度超过 0.2 时, 算法 3 的通信代价显著低于算法 2。值得注意的是, 当干扰时刻集中在数据流的尾部时, 算法 2 和算法 3

的通信代价在某些情况下会有所下降。出现这一现象的原因在于, 此时算法已完成了大部分的统计工作, 进入了相对不受干扰影响的 CMY 过程。因此, 在高干扰密度的尾部时刻, 算法 2 和算法 3 的性能受到的干扰影响反而有所减轻。

4.3 实验结果的综合分析与未来展望

我们通过一系列实验对比了 3 种不同算法在处理受干扰的分布式监控问题时的性能。实验中的干扰次数 Δ 相对于总牌数阈值 N 的比例极小, 几乎不影响整体的概率分布。实验结果表明, 算法 3 (考虑 Δ 次干扰的分布式监控算法) 在面对不同干扰模式时都展现出了较好的稳定性。特别是在干扰集中在数据流尾部时, 算法 3 的通信代价显著低于算法 2。算法 2 在面对较高干扰密度时, 其通信代价的增加尤为显著, 揭示了其对干扰的高度敏感性。

本章的实验结果为理解不同算法在实际应用中的性能提供了重要依据, 并为未来研究和实际部署中的算法优化指明了方向。特别是算法 3, 其在处理干扰时的稳定性和效率, 使其成为分布式监控系统中一个极具前景的候选方案。

结束语 本文探讨了在有概率假设时的分布式监控算法, 通过让算法容许一定程度的干扰, 使算法更为鲁棒, 能够适应多变的实际环境。本文不仅探讨了算法的设计和原理, 还在理论层面证明了其准确性, 并详细分析了通信代价, 证明了在干扰次数较少时, 可达到 $O(K \log \log N)$ 的通信代价, 对于分布式监控问题的研究提供了更多视角和解决思路。

尽管本文在特定的概率假设和有限的干扰情况下展示了算法的有效性, 但该算法仍具有一定的局限性。干扰次数 Δ 的确定就是一个显著的挑战。尽管可以通过历史数据来估算和调整这一参数, 但在实时数据流中直接进行这样的估算仍然极具难度。若 Δ 设置过大, 算法 3 可能会过早地进入 CMY 过程, 未能充分利用预设的概率分布。相反, 若 Δ 设置过小, 算法可能会因局部数据概率分布的波动而快速增加通信代价, 抗干扰能力减弱。未来的研究方向可以探索如何在更高频干扰的环境中设计更鲁棒的分布式监控算法, 以及能够动态调整阈值以适应概率变化的算法。

此外, 对通信代价的讨论也为进一步的研究提供了思路。在实际问题中, 通信代价的定义可能需要更细致的考虑, 例如考虑总的通信比特数而非简单的通信次数, 以更合理、更全面地评估算法的性能。

综上所述, 本研究为分布式监控问题的解决方案提供了有益的见解, 同时也启示了更多可以探索的领域。期待未来的研究能够进一步完善和拓展这一领域的理论框架, 使分布式监控算法更好地适应实际复杂情境。

参考文献

- [1] JUANG P, OKI H, WANG Y, et al. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet [J]. SIGPLAN Not, 2002, 37(10): 96-107.
- [2] MADDEN S R, FRANKLIN M J, HELLERSTEIN J M, et al. TinyDB: an acquisitional query processing system for sensor networks [J]. ACM Transactions on Database Systems, 2005, 30(1): 122-173.
- [3] KERAPURA R, CORMODE G, BRAMAMIRTHAM J.

- Communication-efficient distributed monitoring of thresholded counts [C]//Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. Chicago, IL, USA; Association for Computing Machinery, 2006;289-300.
- [4] CORMODE G, MUTHUKRISHNAN S, YI K. Algorithms for distributed functional monitoring [J]. ACM Trans Algorithms, 2011,7(2): Article 21.
- [5] GANGULY S, GAROFALAKIS M, RASTOGI R. Tracking set-expression cardinalities over continuous update streams [J]. The VLDB Journal, 2004, 13(4): 354-369.
- [6] CORMODE G, GAROFALAKIS M, MUTHUKRISHNAN S, et al. Holistic aggregates in a networked world: distributed tracking of approximate quantiles [C]//Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. Baltimore, Maryland: Association for Computing Machinery, 2005;25-36.
- [7] GIATRAKOS N, DELIGIANNAKIS A, GAROFALAKIS M, et al. Prediction-based geometric monitoring over distributed data streams [C]//Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. Scottsdale, Arizona, USA; Association for Computing Machinery, 2012;265-276.
- [8] BABCOCK B, OLSTON C. Distributed top-k monitoring [C]//Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. San Diego, California; Association for Computing Machinery, 2003;28-39.
- [9] BAR-YOSSEF Z, JAYRAM T S, KUMAR R, et al. Counting Distinct Elements in a Data Stream [C]//Randomization and Approximation Techniques in Computer Science. Berlin, Heidelberg; Springer, 2002.
- [10] CORMODE G, GAROFALAKIS M. Approximate continuous querying over distributed streams [J]. ACM Transactions on Database Systems, 2008, 33(2): Article 9.
- [11] OLSTON C, JIANG J, WIDOM J. Adaptive filters for continuous queries over distributed data streams [C]//Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. San Diego, California; Association for Computing Machinery, 2003;563-574.
- [12] AHMAD Y, BERG B, CETINTEMEL U, et al. Distributed operation in the Borealis stream processing engine [C]//Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. Baltimore, Maryland: Association for Computing Machinery, 2005;882-884.
- [13] CHERNIACK M, BALAKRISHNAN H, BALAZINSKA M, et al. Scalable Distributed Stream Processing [C]//Proceedings of the CIDR. 2003.
- [14] POPA N M, OPRESCU A. A Data-Centric Approach to Distributed Tracing [C]//Proceedings of the 2019 IEEE International Conference on Cloud Computing Technology and Science(Cloud-Com). 2019;11-13.
- [15] CHEN J, DEWITT D J, TIAN F, et al. NiagaraCQ: a scalable continuous query system for Internet databases [C]//Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. Dallas, Texas, USA; Association for Computing Machinery, 2000;379-390.
- [16] MADDEN S, SHAH M, HELLERSTEIN J M, et al. Continuously adaptive continuous queries over streams [C]//Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. Madison, Wisconsin: Association for Computing Machinery, 2002;49-60.
- [17] DOBRA A, GAROFALAKIS M, GEHRKE J, et al. Processing complex aggregate queries over data streams [C]//Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. Madison, Wisconsin; Association for Computing Machinery, 2002;61-72.
- [18] GILBERT A C, KOTIDIS Y, MUTHUKRISHNAN S, et al. How to summarize the universe: dynamic maintenance of quantiles [C]//Proceedings of the 28th International Conference on Very Large Data Bases. Hong Kong, China; VLDB Endowment, 2002;454-465.
- [19] WU H, GAN J, ZHANG R. Learning Based Distributed Tracking [C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Virtual Event, CA, USA; Association for Computing Machinery, 2020;2040-2050.
- [20] HUANG Z, YI K, ZHANG Q. Randomized Algorithms for Tracking Distributed Count, Frequencies, and Ranks [J]. Algorithmica, 2019, 81(6): 2222-2243.
- [21] CHUNG F, LU L. Concentration inequalities and martingale inequalities: a survey [J]. Internet Mathematics, 2006, 3(1): 79-127.



YAN Xinyu, born in 1999, postgraduate. His main research interests include statistics and machine learning.



HUANG Zengfeng, born in 1986, Ph.D., professor, Ph. D supervisor. His main research interests include machine learning and big data algorithms.