量化自动扩缩容器漏洞：云基础设施故障导致资源错配的实证研究

Gijun Park[1],*

[1]Okestro AI Research Center，韩国首尔

arXiv:2601.04659v1 [cs.DC] 8 Jan 2026

摘要。 云环境中的资源自动扩缩容机制依赖准确的性能指标来做出最优的资源配置决策。当基础设施故障（包括硬件故障、网络中断和软件异常）导致这些指标失真时，自动伸缩器可能会系统性地过度或不足配置资源，从而增加运营费用或降低服务可靠性。本文通过受控仿真实验，测量四种常见故障类别对多种实例配置和服务等级目标（SLO）阈值下垂直与水平自动伸缩行为的影响。实验结果表明，存储相关故障会造成最大的成本开销，在水平扩展策略下每月最高增加 258 美元；而路由异常则会持续使自动伸缩器倾向于分配不足的资源。不同扩缩容策略对故障诱发的指标失真的敏感程度差异显著：水平自动伸缩更容易受到瞬态异常的影响，尤其是在阈值边界附近。这些基于实证的发现为设计容错自动扩缩容策略提供了切实可行的建议，帮助区分真实的工作负载波动与故障伪象。

关键词： 云自动扩缩容，基础设施可靠性，故障注入，故障影响量化，成本优化

1 引言

在云计算领域，高效而稳定的资源管理已成为实现可持续运营的一项关键挑战。[1, 2] 云系统的特点是能够弹性扩缩资源，以适应不断变化的工作负载；然而，对于持续消耗大量资源的用户而言，按需付费计费模式有时会导致其成本高于使用同等本地基础设施的成本。 [3]

云自动伸缩器旨在自动扩缩资源，例如虚拟机（VM）或容器，以维持最佳性能并最大限度地减少资源浪费；其采用基于性能指标监控的智能算法，例如资源使用率或请求延迟。 [4]这种动态资源分配使系统能够在工作负载变化时维持既定的性能和可用性。 [5]然而，自动伸缩器的可靠性和有效性高度依赖性能指标的准确性，而云故障可能会对这些指标造成重大影响。 [6]

云故障（例如硬件故障、网络中断或软件问题）可能会导致性能指标出现离群值，进而引发不正确的自动伸缩器行为。[7, 8, 9] 当故障引起的症状导致性能指标无法准确反映实际工作负载规格，且指标值偏低时，自动伸缩器可能会出现欠扩缩容。相反，与故障症状相关的较高性能指标值则可能导致过度扩缩容。这些情况可能引发严重问题，例如资源不足导致服务停机，或资源过度配置造成不必要的额外成本。

尽管人们普遍认为云故障会降低性能，包括导致资源耗尽、请求延迟增加、吞吐量降低以及成本上升，但鲜有研究对特定故障如何影响资源管理提供精确的定量分析。本研究探讨以下关键研究问题。

1. **RQ1**：不同的常见云故障（例如 Disk、软件故障、网络故障）会如何改变自动伸缩器所依赖的 CPU 使用率及其他资源指标?

2. **RQ2**：在不同实例系列中，垂直和水平自动伸缩下，这些变化后的指标会带来多大的成本开销或可靠性风险?

3. **RQ3**：触发自动伸缩的不同阈值（例如资源利用率为 50% 与 85% 的服务等级目标）会如何影响故障导致的资源错配程度及其成本影响?

*通讯作者：Gijun Park (gj.park@okestro.com)

本研究通过科学实验评估并验证了云系统中常见的各种故障对自动伸缩器行为和云使用成本的影响。具体而言，我们分析了每种故障症状与正常状态在性能指标方面的差异。此外，我们还评估了不同类型的故障对垂直和水平自动伸缩某些要素的影响。

2 相关工作

Ravichandiran 等人提出了一种异常检测机制，通过资源行为分析，防止云环境中自动伸缩系统发生经济型可持续性拒绝（EDoS）和资源浪费。 [10]
Moghaddam 等人提出了一种两级自动伸缩框架，将异常检测与水平和垂直扩展策略相结合，以解决云环境中的性能问题。[11]该框架采用基于无监督孤立森林的异常检测方法，分析虚拟机性能指标并预测未来异常，从而能够主动做出扩展决策。

Lalropuia 等人提出了一种基于状态的可用性模型，利用半马尔可夫过程评估 EDoS 攻击下云的稳态可用性。[12]Ahmed 等人提出了一种采用应用级故障注入（ALFI）的实验框架，以研究故障对云服务可扩展性行为的影响。 [13] 他们使用一项基于云的真实软件服务，在两个不同时间点模拟延迟注入，并在 EC2云上开展实验。随后，他们将结果与基准数据进行比较，以研究该云软件服务的可扩展性韧性。Kesavan 等人研究了故障可扩展虚拟化基础设施管理，以提高基础设施即服务（IaaS）管理栈的韧性。 [14]他们提出了一种具备虚拟机推测性复制功能的自动伸缩器：通过在不同物理资源上同时执行多个副本，作为 IaaS 管理栈的扩展，即使部分尝试因故障而失败，也能提高至少一个副本成功的可能性。

虽然以往研究主要聚焦于提升基础设施或服务的主动故障检测能力和韧性，但关于如何系统量化这些中断对自动伸缩结果和成本的影响，所知甚少。承认故障会造成经济影响的现有研究，通常只考察特定故障类型，或依赖无法反映更广泛运营实际情况的局部工作负载。此外，这些研究很少在不同服务等级目标阈值或不同实例类型下，对多种中断进行并列比较，因此难以理解某些故障如何影响资源决策和总体支出，也难以比较不同故障的相对影响。

3 故障感知自动伸缩

自动伸缩算法通常会根据历史观测期内的性能指标（例如 CPU 利用率或每秒请求数 (RPS)），评估资源配置规模及调整是否合理。然而，如果触发自动伸缩流程的数据因故障而不准确，那么无论实际工作负载处于何种状态，自动伸缩算法都可能错误地认为资源配置不足或过量。

云系统中可能发生的主要故障包括网络故障、硬件故障、软件问题、可扩展性与负载问题、安全漏洞以及环境故障。 [15, 16, 17, 18, 19]网络故障包括路由故障，这会干扰数据传输，并可能导致服务不可用。硬件故障包括 CPU 过载、内存耗尽和 I/O 瓶颈，这些问题会降低整个系统的处理能力和性能。软件问题包括实现错误，例如导致意外行为的代码缺陷。安全漏洞包括会耗尽系统资源的拒绝服务（DoS）攻击，以及会损害系统完整性并可能导致数据损坏或未授权访问的恶意软件感染。本研究聚焦于云系统中种类繁多的故障里常见且重要的故障，以增强实验结果的实际适用性。我们根据这些故障对自动伸缩器的影响范围、检测特征和业务相关性，识别并选取了这些故障。

从影响范围来看，网络故障会因系统互联性而跨越多个层级传播：在基础设施层，它们会影响路由和物理连接；在应用层，它们会妨碍 API 调用和微服务交互。 [20] 硬件故障也可能在这些层级间连锁蔓延，影响处理效率并导致更大范围的系统变慢。 [21, 22] 软件问题会扰乱应用功能。诸如 DoS 攻击之类的安全漏洞通常始于基础设施层，通过耗尽网络资源，进而影响平台服务和应用性能。 [23]从检测特征来看，网络故障可能导致延迟增加和丢包，自动伸缩器可能会将其误判为负载激增。 [24]同样，硬件故障可能造成工作负载增加的假象，因为它们会表现出资源利用率升高的症状。 [25] 软件问题可能导致资源泄漏或突然崩溃，进而使自动伸缩器不必要地错误部署资源。 [26]从业务相关性来看，硬件故障和软件问题的恢复复杂度较高，需要投入大量资源并耗费较长时间才能解决；而网络故障会直接影响服务可用性指标，这些指标对于维持服务级别协议（SLA）至关重要。 [27, 28]

因此，本研究聚焦于四类故障：网络故障、硬件故障、软件问题，以及

拒绝服务攻击。这些故障是站点可靠性工程需要重点考虑的因素，因为它们会导致性能指标异常，从而干扰自动伸缩器的正常运行，同时也是事件报告中相当一部分根本原因。 [29]

3.1 故障设置

路由器故障既属于硬件故障，也属于网络故障。当硬件发生故障，或路由基础设施中的网络虚拟化相关异常导致通信延迟异常时，就会发生路由器故障。同一类别中的磁盘瓶颈，是指挂载在虚拟机上的块存储发生故障，导致无法正常处理磁盘 I/O 请求。此类故障通过暂时断开虚拟机与块存储的连接来模拟。重新连接后，累积的 I/O 操作会在处理待处理请求时造成瓶颈。

软件问题涵盖多种故障，从应用程序开发、集成、部署和运维期间的间歇性请求失败，到服务中断。本研究复现了由 API 实现缺陷导致的请求失败。典型的拒绝服务攻击是向目标发送异常流量，耗尽目标的计算资源，从而扰乱正常服务。[30, 31]所研究的攻击包括针对第 7 层（L7）的流量洪泛攻击，以及针对第 4 层（L4）的同步序列号（SYN）泛洪攻击和 UDP 洪泛攻击。

3.2 自动伸缩器

自动伸缩的工作方式如下：如果触发器声明的性能指标超过最佳水平，则视为资源配置不足，并分配额外的计算资源，以提高应用程序的运行稳定性。相反，如果性能指标低于最佳水平，则视为资源过度配置，此时会减少不必要的计算资源以降低成本。自动伸缩大致可分为两种方法。通过直接增加或减少虚拟机的计算资源来管理负载的方法称为垂直自动伸缩；而通过增加副本数量、并行处理工作负载来管理各个虚拟机负载的方法称为水平自动伸缩。

模拟是在垂直和水平自动伸缩场景中进行的，采用 Kubernetes 默认的优化算法。Kubernetes 是云原生架构中常用的容器编排框架。 [32, 33]垂直方法中的最优资源规格按公式 (1) 推导，其中 i 表示一种资源，记为$i \in$ {CPU、内存、磁盘 I/O、网络带宽}， m_i 表示 i每分钟使用量的时间序列数组， SLO 定义为性能指标的最优水平，spec$_i$ 表示 i当前的虚拟机规格，而 optSpec$_i$ 表示 i的最优虚拟机规格。

$$\text{最优规格}_i = \lceil \text{规格}_i \times (\text{最大值}(m_i) \ - (SLO \ - \text{最大值}(m_i))) \rceil \quad (1)$$

在此情境下，服务等级目标以单个值表示，而性能指标则由整个观测期间按固定间隔记录的一系列数值构成。服务等级目标与性能指标的代表值进行比较，该代表值通过最大值聚合得出。 [34]最优实例类型通过网格搜索确定：在所有可用实例类型中，选出计算规格均达到相应资源规格且每小时成本最低的实例类型；这是云服务提供商（CSP）的一项临时条件。

水平方法使用与垂直方法相同的性能指标来确定是否以及在多大程度上进行自动伸缩，但计算方式有所不同。自动伸缩所需的最佳副本数由当前副本数与性能指标目标值和性能指标代表值之比的乘积确定。如果性能指标的代表值高于或低于目标值，则会创建或删除副本，使其更接近目标值。例如，如果代表值超过目标值，则会增加副本，直到达到与该代表值对应的最佳副本数。在这种情况下，最佳副本数的计算方式如式（2）所示。

$$\text{optReplicas} = \max_i(\lceil \text{currentReplicas} \times (\frac{\max(m_i)}{SLO}) \rceil) \quad (2)$$

4 实验

实验环境由两个主要部分组成：服务器组和客户端组。服务器组包含六台虚拟机，分为两个子组：实验组和对照组。实验组包含三台虚拟机，这些虚拟机承载正常工作负载并会发生故障。对照组包含三台仅承载正常工作负载的虚拟机。所有虚拟机均使用 Amazon Web Services (AWS) EC2 部署。客户端组包含一台虚拟机，用于生成工作负载并将其发送至服务器组；另有两台虚拟机用于发起 DoS 攻击。这些虚拟机使用 Google Cloud Platform 的 Compute Engine 服务创建。服务器组和客户端组中的虚拟机在物理上位于不同的数据中心，以确保实验能够反映典型生产环境中的网络影响。自动伸缩器采用逻辑实现，假定服务器资源会立即重新配置为最优规格，具体由公式 1 和 2 确定。任务关键型系统往往采用更为保守的措施来降低运营风险，其中包括考虑故障对自动伸缩器的影响。 [22]因此，该实验反映了以下场景：对于优先考虑

成本效率、关键性较低或使用频率较低的服务，服务等级目标将资源利用率设为 85%；对于强调稳定性的服务级别目标，则设为 50%。实验环境的总体配置以及各组虚拟机的详细规格分别如图 1 和表 1 所示。

在实验中，服务器组中的每台虚拟机先承受正常工作负载 12.5 分钟，随后在正常工作负载下经历第 III 节所述的 5 分钟故障场景。之后，系统仅承受正常工作负载 2.5 分钟。在实际操作中，为了快速响应，决定执行自动扩缩容操作时所采用的回溯时间通常设为 5 分钟。 [35] 然而，在本研究中，为了考虑正常运行、故障开始和故障结束所产生的不同影响，回溯时间设为 15 分钟。通过 Prometheus 以 1 秒为间隔收集服务器组中每台虚拟机的性能指标数据。 [36]正常工作负载由 CloudSuite 4.0 中的网页服务基准测试服务生成。[37]网页服务基准测试通过服务器端模块和客户端模块，全面复现真实社交媒体平台提供的一般功能及用户执行的操作（登录、发帖、添加和删除好友、发送消息等）。它包含处理服务功能的服务器端模块，例如 Web 服务器、数据库服务器和 Memcached，以及生成这些服务功能请求的客户端模块。这些服务器端模块和客户端模块分别作为 Docker 容器部署在服务器组和客户端组中的虚拟机上。

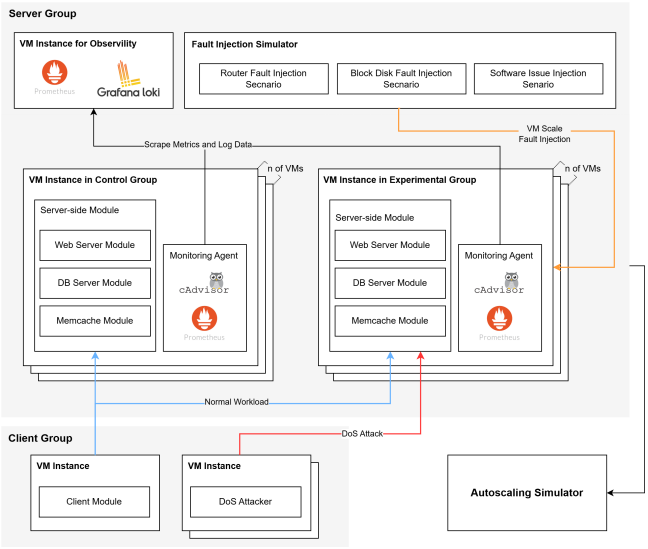

图 1：实验环境架构，包括服务器组（由实验组和对照组组成）、客户端组以及仿真工具。

通过模拟 5 个虚拟用户使用服务 15 分钟，生成正常工作负载。发生拒绝服务攻击时，使用 hping3 工具发起 SYN 和 UDP 洪泛攻击，并使用 MHDDoS 工具实施流量型攻击。[38, 39] hping3 攻击使用 64 字节的窗口大小

表 1：实验虚拟机实例大小的详细规格。

实例类型		CPU	vCPU	内存 (GB)	网络	Cost
系列	Size	性能 (GHz)			（Gbps）	（$/h）
m5	大	3.1	2	8	最大 5	0.104
	超大		4	16		0.208
	2倍超大		8	32		0.416
t3	大		2	8		0.118
	超大		4	16		0.236
	2倍超大		8	32	最大 10	0.482
c5	大	3.3	2	4		0.086
	超大		4	8		0.172
	2倍超大		8	16		0.344

和 120 字节的数据大小；MHDDoS 攻击的线程数为 450，每秒请求数为 150。攻击参数通过实证研究确定：与正常状态下的操作相比，这些参数值导致异常自动伸缩的概率高于 50%；在给定环境中注入故障时，还会造成实际服务中断。使用 AWS 故障注入模拟器（FIS）工具，为所有传入流量在每个虚拟机网络接口上增加 200 毫秒的延迟和抖动，以模拟路由器故障。 [35] 注入的延迟设为普遍接受的 TCP 读取超时默认值。使用 AWS FIS 工具执行卷 I/O 暂停场景，暂时阻断虚拟机上操作系统卷的 I/O，以模拟磁盘故障。在 Web 服务器模块与 Web 服务数据库服务器模块之间的通信中，使用 Docker 混沌测试工具 Pumba 随机引入丢包，以模拟软件问题。 [40] 实现业务逻辑的库和框架采用容错设计，可缓解轻微错误或缺陷，因此故障症状不会轻易体现在性能指标中。因此，软件问题被设定为灾难性故障事件，丢包率为 50%。

5 结果与讨论

5.1 故障对垂直自动伸缩的影响

SYN、UDP 洪泛攻击和大 规格实例的案例表明，在图 2 (a) 所述的 85% 服务等级目标配置下，误差比率始终接近于零。误差比率用于量化故障状态与正常状态之间资源分配决策的偏差，计算方式为

$$E_r = \frac{V_{\text{abnormal}} - V_{\text{normal}}}{V_{\text{normal}}} \times 100 \tag{3}$$

其中， E_r 表示误差比率的百分比，

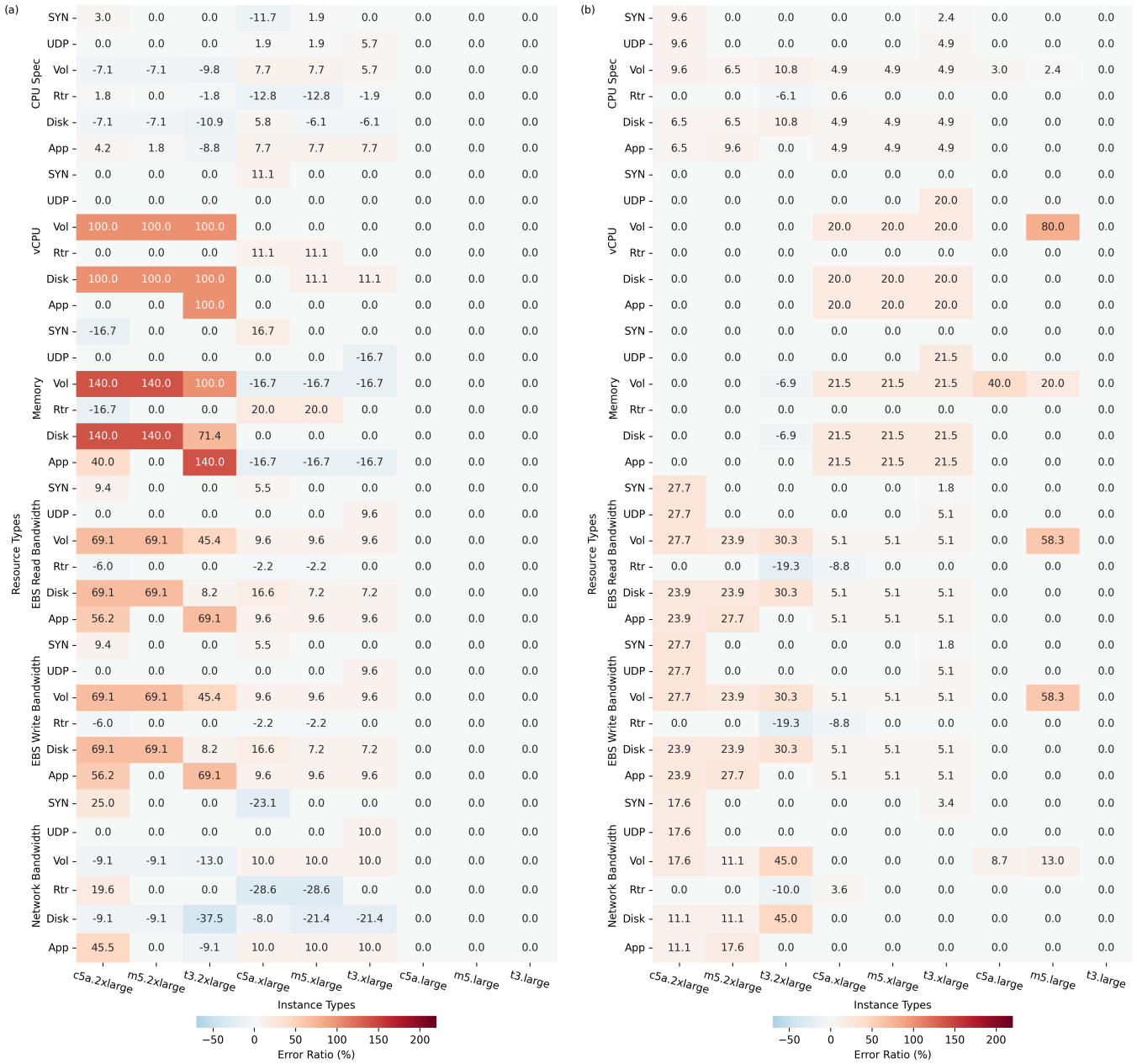

图 2：垂直自动伸缩中，按实例大小比较正常状态与故障状态下最优规格的差距，其中 (a) SLO 85%，(b) SLO 50%。故障包括 SYN 洪泛攻击 (SYN)、UDP 洪泛攻击 (UDP)、流量型攻击 (Vol)、路由器故障 (Rtr)、磁盘故障 (Disk) 和软件问题 (App)。

V_{abnormal} 表示在故障条件下确定的最优资源规格，而 V_{normal} 表示正常运行时的基准规格。正误差比率表示故障导致资源过度配置，负值则表明相对于实际工作负载需求存在资源配置不足。这种稳定性是因为这些攻击会使 CPU 利用率饱和至约 100%，无论实例大小如何，因而自动伸缩决策在正常状态和故障状态下保持一致。相比之下，流量型攻击和磁盘故障会产生显著的正误差比率，尤其是在较大实例类型上。c5a.2xlarge 配置在两种故障类型中的误差比率均超过 140%。这种现象的出现是因为这些故障会导致 CPU 利用率持续处于高位，超过实际工作负载需求，促使自动扩缩器分配不必要的大型实例。同样，软件问题也会导致资源过度配置，c5a.2xlarge 的误差比率为 69.1%。这种资源过度配置是因为软件故障触发的重试逻辑和日志记录流程期间观察到的 CPU 使用率暂时上升，被解读为负载增加所致，

故障被解读为负载增加所致。

路由器故障通常表明存在资源配置不足的倾向，c5a.2xlarge 和 m5.2xlarge 实例的误差比率均为 -16.7%。这是因为路由器故障导致网络延迟，使线程转入 I/O 等待状态，进而令自动伸缩器误以为计算需求有所下降。

如图 2(b) 所示，SLO 场景 50% 在所有故障类型下的影响都较小。流量型攻击和磁盘故障仍会导致资源过度配置，但错误比率有所下降，约为 20%，且对各实例系列的影响范围更窄。具体而言，在较低的 SLO 阈值下，2倍超大规格受到故障的影响更不明显；对于 m5.2xlarge 实例的 EBS 带宽，流量型攻击的错误比率最高仅为 23.9%。这种敏感度下降是因为较低的阈值缩小了正常运行与扩缩容触发点之间的余量。

5.2 故障对水平自动伸缩的影响

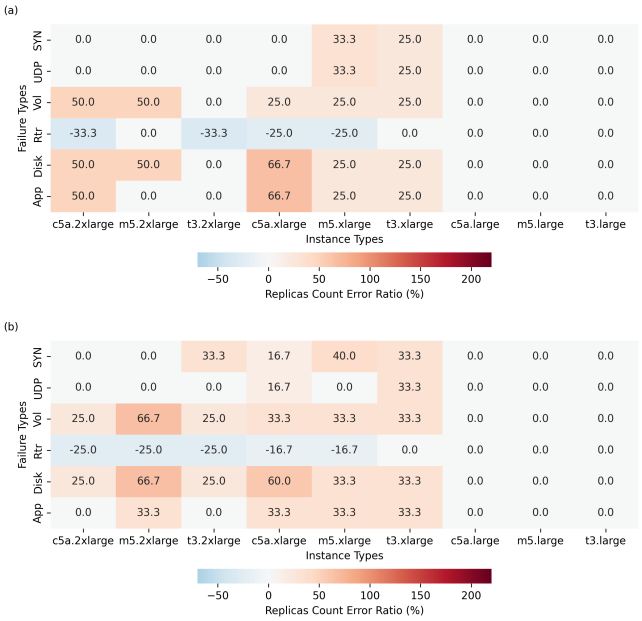

图 3：按实例大小比较水平自动伸缩中正常状态与故障状态之间最优副本数的差距，其中包括 (a) SLO 85% 和 (b) SLO 50%。故障包括 SYN 洪泛攻击（SYN）、UDP 洪泛攻击（UDP）、流量型攻击（Vol）、路由器故障（Rtr）、Disk 故障（Disk）和软件问题（App）。

如图 3(a) 所示，尽管使用相同的指标和阈值，水平自动伸缩呈现出不同于垂直自动伸缩的扩缩容模式。在 85% 服务等级目标下，流量型攻击使 c5a.2xlarge 的误差比率达到 50%，而磁盘故障使 c5a.xlarge 实例的误差比率达到 66.7%。当磁盘故障导致 CPU 利用率接近 100% 时，自动伸缩器会将副本数加倍，进一步加剧资源错配。t3 系列的误差比率保持在 33.3% 或以下，表明突发型实例的特性能够为故障导致的扩缩容错误提供缓冲。

SLO 场景 50% 对故障导致的指标失真表现出更高的敏感性，如图 3 (b) 所示。在 SLO 85% 中影响极小的 SYN 洪泛攻击，如今会使 c5a.xlarge 实例产生 16.7% 的资源过度配置。这种变化是因为较低的阈值与故障导致的 CPU 利用率模式相交，从而触发了不必要的副本增加。软件问题的影响因实例类型而异，其中 超大 规格的实例和 m5.2xlarge 的误差比率为 33.3%，而其他实例则不受影响。

垂直自动伸缩与水平自动伸缩采用不同的优化方法，因此二者产生了差异化影响。垂直自动伸缩将性能指标抽象为实例类型决策，而水平自动伸缩则直接将指标偏差转化为副本数调整。这种直接关系使水平自动伸缩对瞬态指标失真更加敏感，尤其是在接近阈值边界时。

5.3 故障诱发的资源错配所造成的经济影响

量化不同故障条件下自动伸缩错误造成的成本影响后发现，在服务等级目标为 85% 的垂直自动伸缩中，磁盘故障和流量型攻击造成的经济负担最重，每月分别增加约 44 美元和 45 美元的成本，如图4(a)所示。

如图4(b)所示，由于副本倍增，水平自动伸缩显著放大了这些经济影响。在服务等级目标为 85% 时，磁盘故障造成的额外成本最高，每月约为 258 美元；流量型攻击紧随其后，每月约为 241 美元。然而，水平自动伸缩中的路由器故障会使成本每月减少 103 至 138 美元，具体取决于服务等级目标配置。这意味着资源配置严重不足，可能损害服务可用性。

在 50% 的 SLO 阈值下，垂直自动伸缩的成本差异会减小，而水平自动伸缩的成本差异则会增大。这是因为 50% 的较低阈值接近正常的 CPU 利用率。因此，故障导致的较小指标波动更容易越过阈值并触发伸缩操作。于是，垂直自动伸缩可以相对精细地控制资源规格，从而使成本差异较小。相比之下，水平自动伸缩会导致更大的成本差异，因为无论阈值敏感度如何，添加或移除整个副本实例时，指标波动对成本的影响都更大。

成本影响因实例系列和

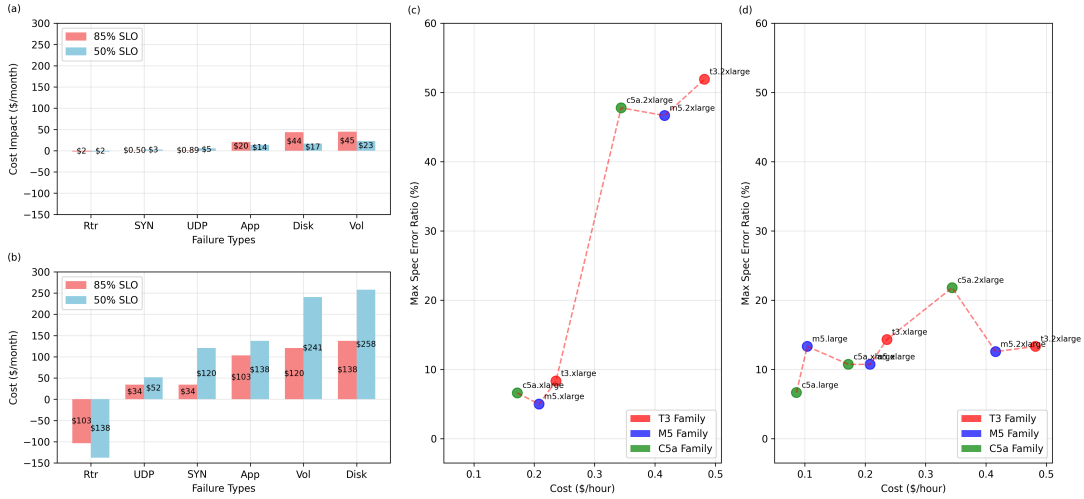

图 4：比较 (a) 垂直自动伸缩和 (b) 水平自动伸缩中，正常状态与故障状态之间月度最优成本差距在不同故障类型下的变化；比较全部故障时，(c) SLO 85% 和 (d) SLO 50% 垂直自动伸缩中，按实例类型划分的故障导致的资源不足风险与成本之比。

规格而异，如图 4(c) 和 (d) 所示。c5a 系列虽然计算性能更优，但在故障导致的过度配置期间，成本始终更高。这种关系表明，性能余量与经济风险之间存在权衡。较小的实例类型通常绝对成本变化较小，但成本相对增幅较高；较大的实例则呈现相反的模式。

5.4 缓解策略与设计考量

实验中发现的扩缩容误差和成本差异，为云服务提供商提供了具体分析托管式自动伸缩服务中漏洞的机会，并可量化其对客户服务质量和成本的潜在影响。云服务用户可以根据观察到的性能模式，为应用选择合适的垂直或水平扩缩容方法，实施故障转移缓解策略，并为容错能力设定服务级别目标。

更具体地说，对于磁盘故障，其特征是 I/O 任务中断、CPU 利用率接近 100% 等持续性症状。在允许的停机时间内逐步触发 I/O 监控，有助于区分正常工作负载增加与暂时性的 I/O 瓶颈。尤其是在对计算优化型或加速器实例等高级虚拟机进行激进扩缩容时，应格外谨慎，因为这可能导致成本过高。对于 SYN 和 UDP 洪泛攻击等 DoS 攻击，可以通过配置触发器来提升自动伸缩器的容错能力：当多个受故障影响较小的互补指标同时超过各自的预定义阈值时触发，例如在故障期间呈负相关的 CPU 利用率和网络吞吐量，而非仅依赖默认指标。对于路由器故障，研究发现其往往会导致各实例系列的资源配置不足。采用能够监测网络延迟阈值并将其纳入触发条件的延迟感知型扩缩容策略，有助于维持延迟敏感型应用的服务可靠性。

在架构决策方面，服务可靠性错误预算较为充裕的组织，或许可以优先考虑垂直自动伸缩的成本效益。虽然垂直自动伸缩难以应对工作负载的显著变化，响应能力也不及水平自动伸缩，但其在正常状态与故障状态下的行为变化较小。这种一致性有助于防止故障触发非预期的自动伸缩操作。

6 结论

本研究通过系统的基于仿真的实验，全面量化评估了常见云故障对自动伸缩机制的影响。通过分析垂直自动伸缩和水平自动伸缩在各种故障场景下的行为，我们证明故障会显著扭曲性能指标，进而导致资源错配，并造成可量化的经济后果。

我们的实验结果揭示了三项发现。首先，磁盘故障造成的经济负担最为严重，在水平自动伸缩场景下，每月额外成本最高可达 258 美元；而网络故障可能误导自动伸缩器，导致系统性资源配置不足。其次，故障的影响因实例系列和自动扩缩容策略而异，其中计算优化型实例更容易受到故障诱发的资源配置不足影响。

第三，SLO 阈值从根本上影响资源错配的程度：较低的阈值对指标失真更敏感，但绝对成本影响较小。这些发现为将故障感知纳入自动伸缩决策算法奠定了基准，使自动伸缩器能够通过多指标验证和非对称伸缩阈值，区分由工作负载驱动的资源需求与由故障诱发的资源需求。因此，云服务提供商和用户都能采用自动扩缩容策略，在故障条件下改善服务连续性，同时最大限度地降低成本与风险之间的权衡。

然而，尽管用作仿真环境的 CloudSuite 4.0 Web 服务基准测试复现了真实应用的相同功能和工作负载，但与仿真环境相比，生产环境中的流量和资源竞争可能会有很大的波动，这是由整体云架构的复杂性及各种外部变量造成的。该故障注入场景体现了典型云使用环境中的主要故障；然而，在某些特殊环境中，未涵盖的低频但影响重大的问题可能更为重要，例如级联故障和多区域中断。未来，我们计划基于本研究的发现，分析多种应用类型、架构模式和工作负载特征下更多复合故障，并考察这些故障及其导致的性能下降对包括自动伸缩在内的高可用配置的影响。

参考文献

[1] S. Aslam、S. u. Islam、A. Khan、M. Ahmed、A. Akhundzada 和 M. K. Khan，《面向云资源管理的信息收集中心型技术：分类法、分析与挑战》，第 100 卷，第 80–94 页。 [在线]。可访问： https://www.sciencedirect.com/science/article/pii/S1084804517303545[2] S. Mustafa、B. Nazir、A. Hayat、A. u. R. Khan 和 S. A. Madani，《云计算中的资源管理：分类法、前景与挑战》，第 47 卷，第 186–203 页。[在线]。可访问：https://www.sciencedirect.com/science/article/pii/S004579061500275X[3] S. Mireslami、L. Rakai、M. Wang 和 B. H. Far，《考虑需求不确定性的动态云资源分配》，第 9 卷，第 3 期，第 981–994 页，会议名称：IEEE Transactions on Cloud Computing。 [在线]。可访问： https://ieeexplore.ieee.org/document/8633415[4]M. A. Tamiru、J. Tordsson、E. Elmroth 和 G. Pierre，《云中的 Kubernetes 集群自动扩缩器实验评估》，载于*2020 IEEE* 云计算技术与科学国际会议*(CloudCom)*，第 17–24 页，国际标准连续出版物编号：2330-2186。 [在线]。可访问：https://ieeexplore.ieee.org/document/9407312[5] P. T. Endo、M. Rodrigues、G. E. Gonçalves、J. Kelner、D. H. Sadok 和 C. Curescu，《云中的高可用性：系统综述与研究挑战》，第 5 卷，第 1 期，第 16 页。 [在线]。可访问：https://doi.org/10.1186/s13677-016-0066-8

[6]S. Taherizadeh 和 M. Grobelnik，《Kubernetes 自动扩缩器对计算密集型微服务原生云应用的关键影响因素》，第 140 卷，第 102734 页。 [在线]。可访问：https://www.sciencedirect.com/science/article/pii/S0965997819304375[7] N. Herbst、A. Bauer、S. Kounev、G. Oikonomou、E. V. Eyk、G. Kousiouris、A. Evangelinou、R. Krebs、T. Brecht、C. L. Abad 和 A. Iosup，《量化云性能与可信性：分类法、度量设计及新兴挑战》，第 3 卷，第 4 期，第 19:1–19:36 页。 [在线]。可访问：https://doi.org/10.1145/3236332

[8] X. Xu、H. Li，《云自动扩缩容机制中的性能攻击缓解》，第 6 卷，第 3 期，第 369–376 页。 [在线]。可访问：https://www.sciencedirect.com/science/article/pii/S2352864819301440

[9] V. Kalavri、J. Liagouris、M. Hoffmann、D. Dimitrova、M. Forshaw 和 T. Roscoe，《三步即可：分布式流数据流的快速、准确自动扩缩容决策》，第 783–798 页。 [在线]。可访问：https: //www.usenix.org/conference/osdi18/presentation/kalavri

[10] R. Ravichandiran、H. Bannazadeh 和 A. Leon-Garcia，《用于自动伸缩系统的资源行为分析异常检测》，载于 *2018*年第四届*IEEE*网络软化会议及研讨会 *(NetSoft)*，第 192–196 页。 [在线]。可访问：https://ieeexplore.ieee.org/document/8460025

[11]S. K. Moghaddam、R. Buyya 和 K. Ramamohanarao，《ACAS：面向云的基于异常的因果感知自动伸缩框架》，第 126 卷，第 107–120 页。 [在线]。可访问：https://www.sciencedirect.com/science/article/pii/S0743731518309080

[12] K. Haribabu（原名 Gupta），《经济型可持续性拒绝（EDoS）攻击下云计算的可用性与可靠性分析：一种半马尔可夫方法》，第 24 卷，第 3 期，第 2177–2191 页。 [在线]。可访问： https://doi.org/10.1007/s10586-021-03257-9

[13] A. Andrade、A. Sidi，《使用应用级故障注入的云软件服务可扩展性韧性框架》，第 11 卷，第 1 期，第 1 页。 [在线]。获取地址：https://doi.org/10.1186/s13677-021-00277-z

[14] M. Kesavan、A. Gavrilovska、K. Schwan，《故障感知虚拟化基础设施管理》，载于 *2017*年*IEEE*第*37*届分布式计算系统国际会议（*ICDCS*），第 252–263页，国际标准连续出版物编号：1063-6927。 [在线]。网址： https://ieeexplore.ieee.org/document/7979972

[15]M. Yousif，“云计算可靠性——故障也是一种选择”，第5卷，第3期，第4–5页，会议名称：IEEE Cloud Computing。 [在线]。可访问：https: //ieeexplore.ieee.org/document/8383673[16] P. Garraghan、R. Yang、Z. Wen、A. Romanovsky、J. Xu、R. Buyya 和 R. Ranjan，“涌现故障：重新思考大规模云可靠性”，第5卷，第5期，第12–21页，会议名称：IEEE Cloud Computing。 [在线]。可访问：https://ieeexplore.ieee.org/document/8497007[17] H. Wang、H. Shen 和 Z. Li，“云数据中心级联故障韧性应对方法”，载于 *2018*年*IEEE*第*38*届分布式计算国际会议

系统（*ICDCS*），第 706–717，国际标准连续出版物编号：2575-8411。[在线]。可访问：https://ieeexplore.ieee.org/document/8416337

[18]Y. Liu、M. Ma、P. Zhao、C. Bansal、S. Rajmohan、L. Qingwei 和 D. Zhang，“Early Bird：通过早期故障预测确保云系统的可靠性”，发表于*2024*年国际软件可靠性工程研讨会，2024年10月。[在线]。可访问：
https://www.microsoft.com/en-us/research/publication/early-bird-ensuring-reliability-of-cloud-systems-through-early-fail障预测/

[19] T. N. Tengku Asmawi, A. Ismail, and J. Shen，《基于传统机器学习和深度学习的云故障预测》，云计算期刊，第11卷第1期，第47页，2022年9月。[在线]。参见：https://doi.org/10.1186/s13677-022-00327-0

[20] D. R. Mathews, M. Verma, P. Aggarwal, 和 J. Lakshmi，《云堆栈中的多层故障传播与分析洞见》。IEEE计算机学会，2021年9月，第714–716页。[在线]。可访问：https://www.computer.org/csdl/proceedings-article/cloud/2021/006000a714/1ymJ0B2THxe

[21] N. Chen, Q. Zhong, Y. Liu, W. Liu, L. Bai, and L. Hu，《基于云环境中时空相关性的容器级联故障检测》，云计算期刊，第 12 卷，第 1 期，第 59 页，2023年4月。[在线]。可访问：https://doi.org/10.1186/s13677-023-00438-2

[22] G. Dong、Y. Hua、Y. Zhang、Z. Chen 和 M. Chen，《理解并检测云系统中的慢故障硬件故障漏洞》，2025年，第 1127–1142 页。[在线]。可访问：https://www.usenix.org/conference/atc25/presentation/dong

[23] N. Agrawal, S. Tapaswi，《云计算环境中针对 DDoS攻击的防御机制：最新进展与研究挑战》，*IEEE*通信调查与教程，第 21 卷，第 4 期，第 3769–3795 页，2019 年。[在线]。获取地址：https://ieeexplore.ieee.org/document/8794618

[24] C. Wang、Y. Lou、X. Peng、J. Liu 和 B. Zou，《挖掘资源操作知识以支持资源泄漏检测》，载于第 *31* 届 *ACM* 欧洲软件工程大会与软件工程基础研讨会联合会议论文集，丛书：ESEC/FSE 2023。计算机协会，第 986–998 页。[在线]。获取地址：https://doi.org/10.1145/3611643.3616315

[25]A. Strikos。解析 CPU 速度：利用率如何影响性能。[在线]。获取地址：https://github.blog/engineering/architecture-optimization/breaking-down-cpu-speed-how-utilization-impacts-performance/

[26] O. A. Akanbi、A. Aljaedi、X. Zhou 和 A. R. Alharbi，《软件定义网络中分布式控制器架构的快速故障切换技术》，*IEEE Access*，第 7 卷，第 160 718–160 737 页，2019 年。

[27] E. Bauer 和 R. Adams，《硬件可靠性、虚拟化与服务可用性》，载于云计算的可靠性与可用性。纽约州纽约市：IEEE，2012 年，第 116–131 页。[在线]。可访问：https://ieeexplore.ieee.org/abstract/document/6305406

[28]Y. Mo 和 L. Xing，《面向云计算系统的资源可用性高效分析，以减少 SLA 违约》，*IEEETransactions on Dependable and Secure Computing*，第 19，第 6，页 3699–3710，1月 2022. [在线]。可访问：https://ieeexplore.ieee.org/document/9517012

[29] L. Fiondella, S. Gokhale, V. Mendiratta, 和 A. Casale，“云事件数据：实证分析”，载于*2013*年*IEEE*云工程国际会议 *(IC2E)*，第 241–249 页。[在线]。可访问：https://ieeexplore.ieee.org/document/6529290

[30]Q. Yan 和 F. R. Yu，“云计算环境下软件定义网络中的分布式拒绝服务攻击”，第 53 卷，第 4 期，第 52–59 页，会议名称：IEEE通信杂志。[在线]。可访问：https: //ieeexplore.ieee.org/document/7081075

[31] M. Marsa-Maestre, R. Rathore, 和 J. Salim，《物联网中的分布式拒绝服务攻击及其防御措施：综述》，第 76 卷，第 7 期，第 5320–5363 页。[在线]。可访问：https://doi.org/10.1007/s11227-019-02945-z

[32]“水平 Pod 自动扩缩”，章节：文档。[在线]。可访问：https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/

[33] “自动伸缩器/垂直 Pod 自动扩缩器（主分支）· Kubernetes/自动伸缩器。”[在线]。可访问：
https://github.com/kubernetes/autoscaler/tree/master/vertical-pod-autoscaler

[34] H. Qiu, W. Mao, C. Wang, H. Franke, A. Youssef, Z. T. Kalbarczyk、T. Başar 和 R. K. Iyer，“{AWARE}：利用强化学习在生产云系统中实现工作负载自动扩缩”，第 387–402 页。[在线]。可用：https: //www.usenix.org/conference/atc23/presentation/qiu-haoran

[35] “扩缩计划的最佳实践 - AWS Auto Scaling。”[在线]。可用：https://docs.aws.amazon.com/autoscaling/plans/userguide/best-practices-for-scaling-plans.html[36]“parsa-epfl/cloudsuite”，原始日期：2016-01- 04T11:19:22Z。[在线]。可用：https://github.com/ parsa-epfl/cloudsuite

[37]“prometheus/prometheus”，原始日期：2012-11-24T11:14:12Z。[在线]。可用：https://github.com/prometheus/prometheus[38]S. Sanfilippo，“antirez/hping”，2025 年 3 月，原始日期：2012-06-13T17:41:54Z。[在线]。可用：https: //github.com/antirez/hping[39]“MatrixTM/MHDDoS”，原始日期：2020-10- 13T19:02:46Z。[在线]。可用：https://github.com/ MatrixTM/MHDDoS[40]A. Ledenev，“alexei-led/pumba”，原始日期：2016-03-22T14:20:27Z。[在线]。可用：https://github.com/alexei-led/pumba