论文分享 Saguaro：一种支持边缘计算的分层联盟链

简述

边缘计算通过将计算能力推向网络边缘，显著降低延迟并提升实时性。然而，设备间的跨域通信复杂性随之增加，涉及多层服务器架构的问题，例如网络延迟和数据安全等，亟需有效的解决方案。优化跨域数据传输与事务处理因此成为边缘计算网络中的核心议题。本文提出了 Saguaro，一种专为边缘计算网络设计的许可链系统。Saguaro 采用层次化架构优化跨域应用，减少广域通信的开销，同时引入协调者协议和乐观协议以低延迟处理跨域事务。其分层结构通过数据聚合有效降低高层域的负载。此外，Saguaro 支持移动设备的事务处理，减少了对跨域共识协议的依赖。实验结果表明，该系统在处理跨域和移动事务方面具有显著的可扩展性和高效性。

作者

本文是多机构合作研究的成果，主要作者之一 Mohammad Javad Amiri 拥有丰富的学术背景。他曾在宾夕法尼亚大学担任博士后研究员，并在加州大学圣塔芭芭拉分校获得计算机科学博士学位，目前任职于美国石溪大学计算机科学系，担任助理教授。Amiri 的研究主要聚焦于数据管理与分布式系统，特别是分布式事务处理、共识协议和区块链技术。他的研究成果多次发表于 VLDB、NSDI 和 SIGMOD 等顶级学术会议。另一位重要作者 Boon Thau Loo 是宾夕法尼亚大学计算机与信息科学系的 RCA 教授，同时担任工程学院教育与全球事务高级副院长，负责管理近 5000 名博士生和专业硕士生。他领导 NetDB@Penn 研究团队，并积极参与分布式系统实验室及 Penn 数据库组的研究。此外，Boon 还是两家公司的联合创始人：Termaxia，一家专注于软件定义的大数据存储技术的公司，以及 Netsil一家提供云性能分析的企业。

背景

随着边缘计算、区块链等新兴分布式技术的迅速发展，分布式应用正从传统的数据库架构向更加开放和公众化的应用模式转变。边缘计算通过将计算资源部署到网络边缘，使数据处理更加靠近设备端，从而显著降低了延迟并提高了实时响应能力。然而，这也带来了跨域通信和数据安全等复杂问题，尤其是在设备和服务器层级较多的情况下，如何高效且安全地进行数据传输和事务处理，成为了亟待解决的核心挑战。传统的分布式应用通常依赖中心化的数据处理架构，但在边缘计算环境下，这种架构面临诸多局限性。例如，边缘设备通常分布在多个地理区域，并需要在不同的域之间频繁通信。这种跨域通信不仅增加了网络延迟，还可能涉及多个层次的服务器，导致事务处理效率低下。此外，高层设备往往难以实时追踪各个边缘设备的具体状态，从而无法进行有效的应用级处理。而且，边缘设备的高度移动性对跨域事务处理提出了更高的要求，因为设备可能随时移出本地域，参与远程域的操作，进一步增加了事务一致性和数据安全的挑战。

为了解决这些问题，本文提出了 Saguaro，一个专为边缘计算环境设计的联盟链系统。Saguaro 通过充分利用边缘计算的层次化架构，优化了跨域通信，特别是在多个地理区域之间。它设计了一种基于协调器和乐观估计的共识事务处理协议，有效降低了跨域事务的时延；同时，采用按轮进行的数据聚合协议，使云服务器能够定期收集和了解整个网络下的区块链状态。此外，Saguaro 还设计了一种支持移动设备的共识协议模块，允许边缘设备在不同地域发起事务，

而不破坏数据一致性。通过这些创新，Saguaro 提供了一种高效、安全、可扩展的解决方案，尤其在共享出行、微支付等典型应用场景中，能够实现更加高效的跨域事务处理和数据聚合。

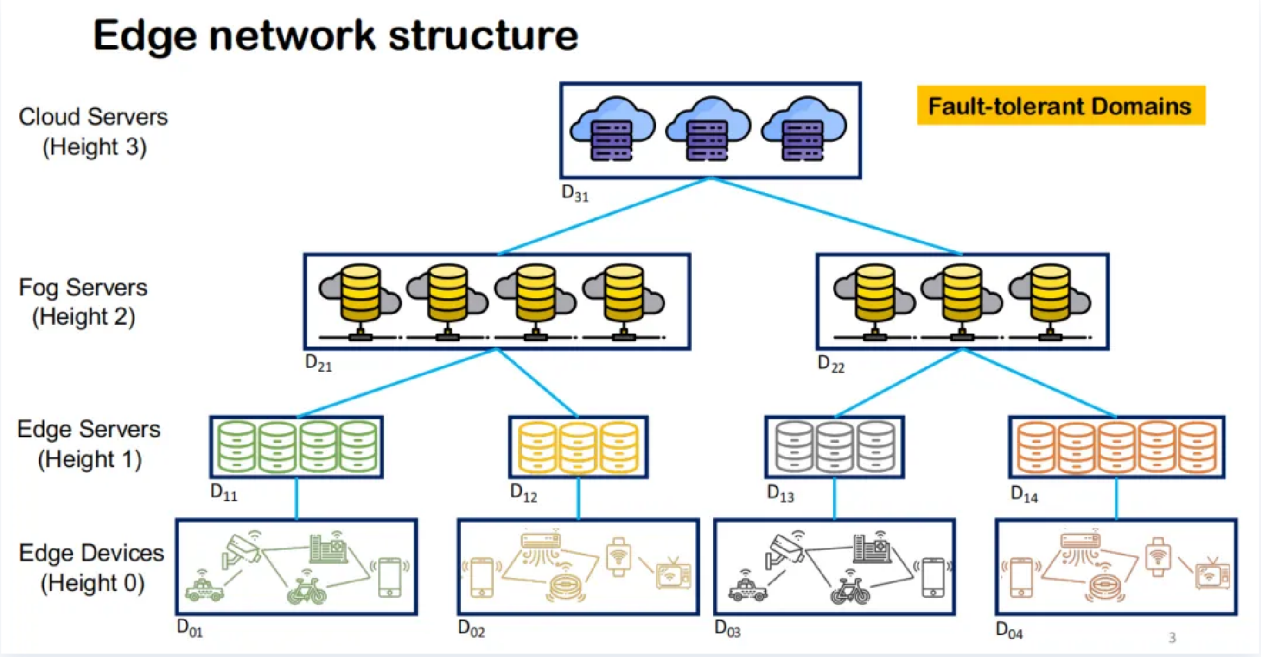
贡献

事务处理共识协议：Saguaro 提供了两种事务处理共识协议，通过基于协调器的共识协议和基于乐观估计的共识协议，充分利用边缘计算的分层架构。这些协议能够有效支持在同一容错域内以及跨容错域的事务处理。

数据聚合协议：Saguaro 提供了一种数据聚合协议，同样依托边缘计算的分层特点。在该协议中，低层域负责执行和维护线性账本，而高层域则仅维护子域的有向无环图（DAG）结构的汇总视图，从而优化了数据的管理和处理效率。

支持移动设备：Saguaro 设计支持移动设备，使得边缘设备即便在不同域之间迁移，也能高效发起并处理跨域事务，即使设备远离初始本地域，仍能确保事务的一致性和高效性。

系统模型



假设

系统运行于一个分布式的边缘计算网络中，参与的各个节点由已知且身份明确的参与者组成，但可能存在不可信的情况。系统采用部分同步通信模型，假设存在全局稳定时间（GST）。在 GST 之后，所有正确节点之间的消息能够在有限时间内传递。节点通过双向点对点通信通道进行交互。系统假定可能会遭遇拜占庭攻击，攻击者可以控制和协调部分节点以干扰系统运行，但无法突破标准加密假设。

模型

Saguaro 网络由层次化的容错域（Domain）构成，涵盖边缘设备、边缘服务器、雾计算服务器及云服务器等多个层级。每个容错域包含多个节点（根域除外），以实现容错性。系统中，节点遵循不同的故障模型，包括崩溃故障模型和拜占庭故障模型。在崩溃故障模型中，节点可能会停止运

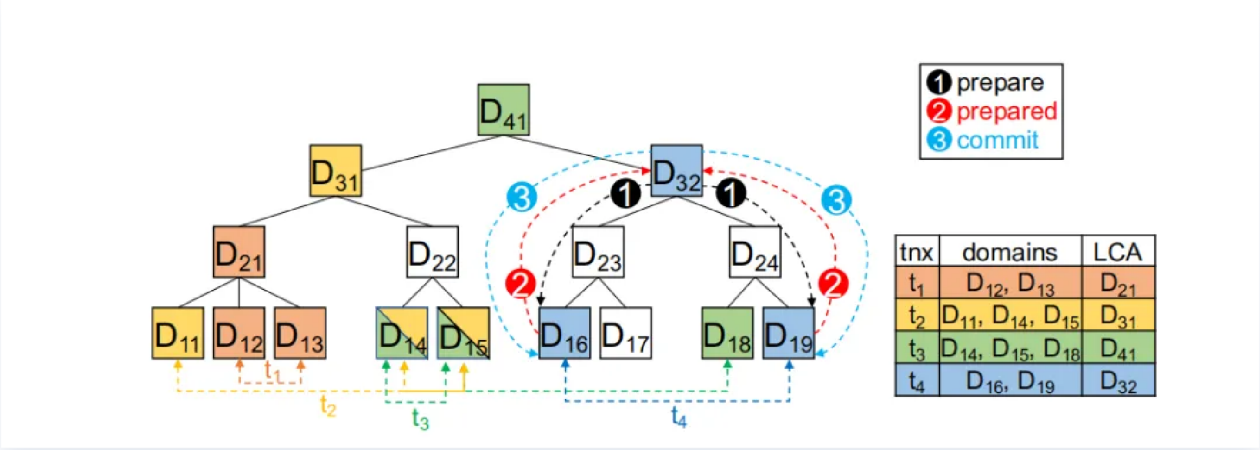
行并重新启动；在拜占庭故障模型中，节点可能表现出任意恶意行为。根据故障模型，每个容错域分别采用崩溃容错协议（如 Paxos）或拜占庭容错协议（如 PBFT）进行账本复制，从而保障系统的安全性和一致性。

规范说明

1. “域”指由若干服务器节点组成的容错域。Height为1 的域被称为“边缘域”，更高层级的域分别为“雾域”和“云域”，以此类推。
2. 跨域事务涉及多个域，这些域的集合被称为“涉及域”。涉及域的最近公共祖先域简称为 LCA 域。
3. 一个域的主节点负责协调该域内部协议运行。在崩溃容错模型（CFT）中，主节点是领导节点；在拜占庭容错模型（BFT）中，主节点是发起共识的节点。一个域向另一个域发送消息，指的是前者的主节点将消息多播到后者的所有节点。
4. 任何不符合协议的行为被正常节点发现后都将触发换届机制，此过程不再在后文单独说明。
5. 下文中，事务或事务均指一个数据库下的有效读写请求

协议内容

基于协调器的跨域共识协议



tnx	domains	LCA
t_1	D_{12}, D_{13}	D_{21}
t_2	D_{11}, D_{14}, D_{15}	D_{31}
t_3	D_{14}, D_{15}, D_{18}	D_{41}
t_4	D_{16}, D_{19}	D_{32}

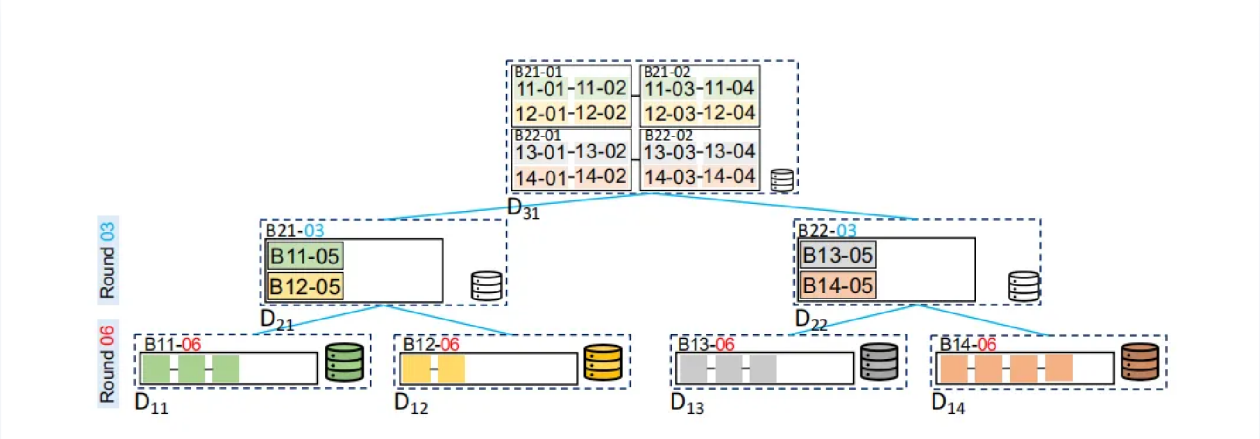
基于协调器的共识协议的核心思想是利用 LCA 域作为跨域事务的排序决策者。当跨域事务发生时，边缘域将该事务发送至 LCA 域，请求其进行排序决策。LCA 域的主节点负责做出排序决定，并通知所有涉及域按照该顺序处理和执行事务。

具体协议步骤如下：

- **Prepare 阶段**
  当边缘域收到一个跨域事务时，其将事务转发给 LCA 域。LCA 域的主节点在接收到该事务后，首先检查冲突：若当前正在处理的事务与该事务的涉及域存在交集，LCA 域的主节点将推迟处理该事务，直到当前正在处理的事务提交完成，以保证交集域中的事务账本一致；若不存在冲突，LCA 域的主节点会为该事务分配一个序列号，并将事务的序列号、事务摘要以及事务本身一起作为 **Prepare 信息** 发送给所有涉及域。

- **Prepared 阶段**
  当涉及域的主节点收到 **Prepare 信息** 后，其检查冲突。若不存在冲突，该主节点将为该事务分配序列号，并在该域内发起对该事务共识。一旦共识达成，主节点会将共识结果作为 **Prepared 信息** 通过签名形式发送回 LCA 域。
- **Commit 阶段**
  当 LCA 域的主节点接收到来自涉及域的有效 **Prepared 信息** 后，它将在域内发起对该事务的共识，并最终生成一个包含所有涉及域序列号的 **Commit 信息**。如果有任何涉及域未同过对该事务的共识，则LCA 域会发送 **Abort 信息**。一旦所有涉及域的节点收到**Commit 信息**，则视为事务已提交，各域按顺序执行该事务；如果收到 **Abort 信息**，则视为事务被抛弃，相关节点将丢弃该事务。
- **Execution 阶段**
  节点在收到有效的 **Commit 信息** 后，将根据应用的设计执行相应的事务操作。

Lazy形式的数据聚合共识协议

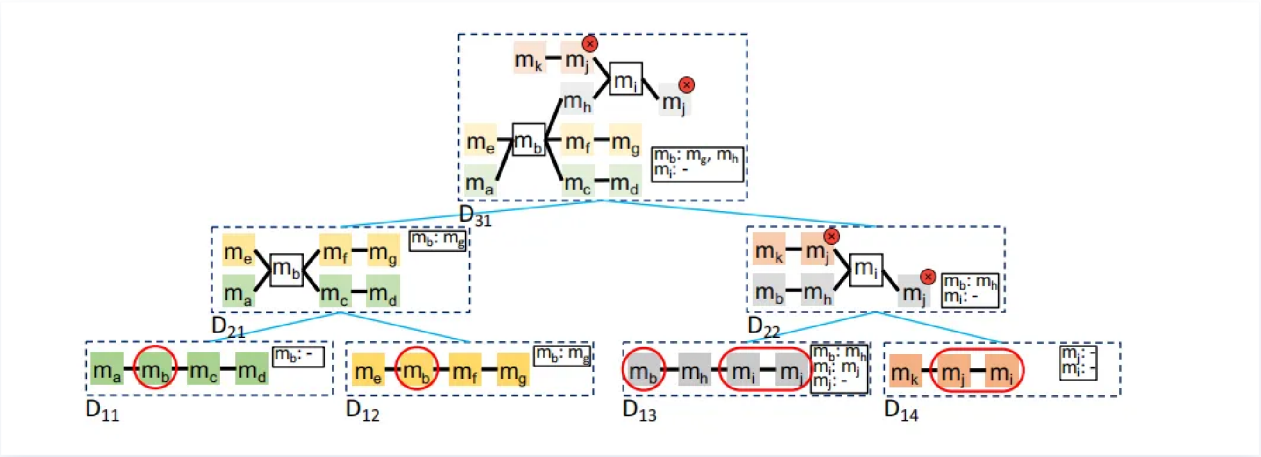


为了使得更高层次的服务器能够及时了解事务数据信息，Saguaro 设计了一套数据聚合共识协议，用于将数据逐层向上传递。为了减少网络开销，该数据传递以 **Lazy** 方式进行。

以边缘域为例，在每个预定时间长度的回合结束后，每个边缘域的主节点会在 **Proposal 信息** 中插入一个 **截止标签 Cut**，标记该回合的结束。所有在截止标签之前的事务将被打包为一个区块信息。接着，边缘域对该区块信息进行共识，并将共识完成后的区块消息发送至父级域。如果某个域在该回合内没有收到任何事务，它将发送一个空的区块消息。

根据子域的故障模型，区块消息将由主节点（在崩溃故障模型下）签名，或者由至少 **2f + 1** 个节点签名（在拜占庭故障模型下）。在更高层级的域中，域内的节点对从子域接收到的区块消息进行共识，完成共识的区块消息将进一步上传至父级域。如果父级域在预定时间内未收到来自子域的区块消息，它将向子域发送查询消息，并启动子域的换届机制。由于高度为2及以上的域可能有多个子节点，每个域将从多个子域接收区块消息，并对回合内收到的所有事务进行排序。如果不同子域的事务之间没有依赖关系，则任意的事务顺序都是可行的。然而，对于跨域事务，必须确保每笔事务仅在父域账本中附加一次，从而使得生成的账本形成一个有向无环图（DAG），以捕获事务之间的顺序依赖关系。对于同一层级的多个域，它们的回合时间间隔是相同的；高层域可以采用较长的时间间隔，以减少通信开销。最终，根域的账本将包含系统中处理的所有事务。

基于乐观的跨域共识协议



除了基于协调器的跨域共识协议，Saguaro 还提供了一套基于乐观的共识协议。该协议的核心思想是，每个涉及域都假设其他涉及域也会按照一致的顺序提交事务，因此可以提前进行预提交和执行，通过 **Lazy** 形式的数据聚合共识协议传播事务状态后，由LCA 域最终进行一致性检查，识别是否存在排序冲突。其实际是受到了前者基于协调器的跨域共识协议和数据聚合协议共同启发的（因为实际数据聚合协议会将区块信息逐层上传，一定会经过LCA域，因此与其在跨域时将事务交给LCA域协调，不如Lazy下来，在传播阶段让LCA域来处理冲突检测）。

具体步骤如下：

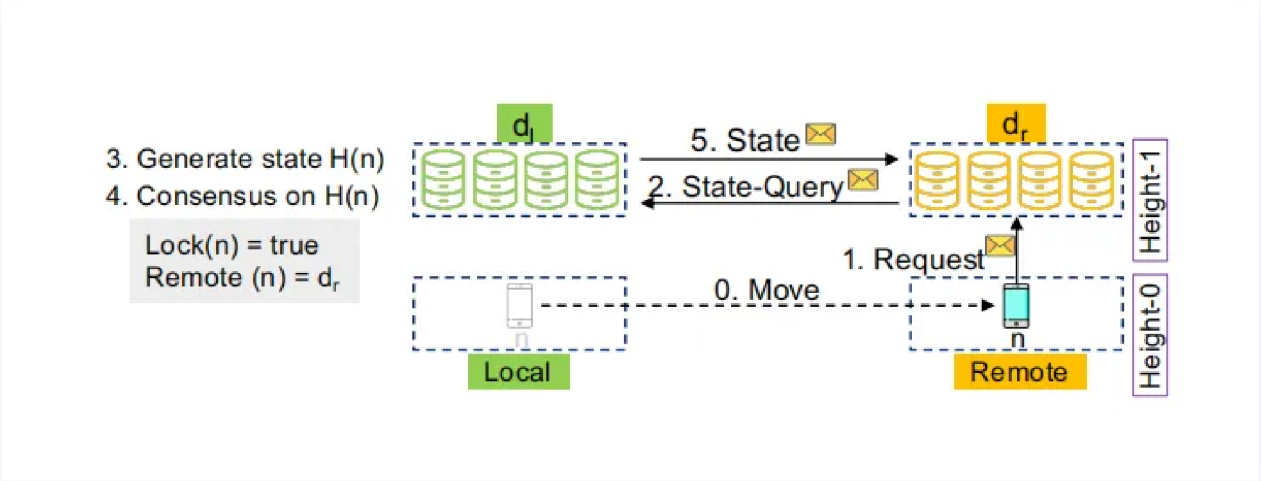
- **回合内的共识阶段**

当某个边缘域收到一条跨域事务时，其将该事务发送至所有涉及域。每个涉及域使用内部共识协议提交并执行该事务，同时维持一个依赖表，记录所有直接或间接依赖该事务的后续记录。

- **回合结束后的传播阶段**

在数据聚合传播过程中，最终 LCA 域将接收到所有涉及域的区块信息，并据此判断事务排序是否一致。如果某个事务未能通过一致性检查，LCA 域将把该事务以及所有直接或间接依赖该事务的事务标记为 **Aborted**，并将相关证明发送至涉及域。所有涉及域需回滚该事务及其所有依赖事务。

移动共识协议



当设备从本地域移动到远程域时，远程域无法在没有本地状态信息的情况下进行事务处理。为了解决这一挑战，Saguaro 提出了一种 **移动共识协议**，使得本地域可以将移动设备的状态信息共享至远程域，从而允许远程域处理由该设备发起的事务。该协议的核心思想是，本地域通过一轮时间将设备的最新信息传递给远程域，确保远程域能够执行事务。

具体来说，每个域都会维护以下几个重要信息：

- 设备的状态 `H`
- 设备的移动状态锁位（lock bit），用于跟踪设备的状态变化（如果设备发起的事务位于远程域，锁位会被设置为 `FALSE`，表示本地域的状态已过时）
- `remote` 变量，指向拥有设备最新事务记录的远程域

当设备在远程域发起事务时，接收该事务的远程域将向设备所属的本地域发送 **State-Query 信息**，请求设备的最新状态信息。若本地域收到有效的 **State-Query 信息**，其将检查设备的锁位状态：

- **若锁位为** `TRUE`，表示本地域的状态是最新的。本地域会调用 `GENERATESTATE` 函数生成设备的最新状态，并在本地域进行共识。一旦共识完成，本地域将通过签名消息将生成的状态信息返回给远程域。
- **若锁位为** `FALSE`，表示本地域的状态已经过时。此时，本地域会通过 `GETSTATE` 函数向 `remote` 变量指向的远程域请求设备的最新状态。获得更新状态后，本地域将与远程域一起完成共识，更新区块链账本，并将更新后的状态信息返回给远程域。

当设备从一个远程域迁移到另一个远程域时，本地域将充当中介角色，通过向新域传递最新状态，确保设备能够在新域继续进行事务处理。如果设备返回本地域，则本地域会更新账本并处理相应的事务。

实验

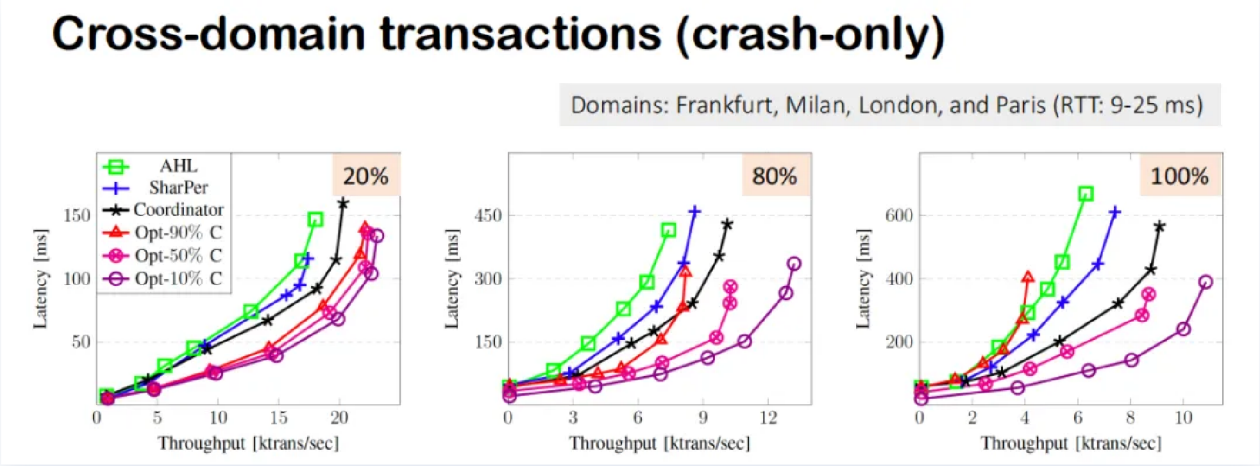
实验设置

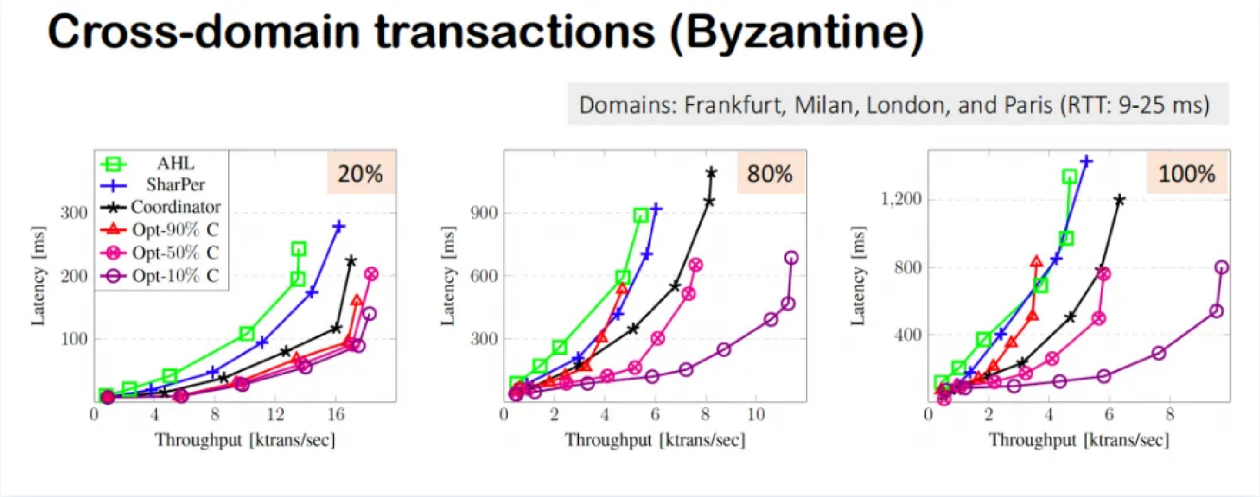
Saguaro 的实验评估主要关注以下几个因素对系统性能的影响：广域事务、跨域事务、以及移动事务对系统性能的影响。实验采用四层满二叉树的边缘计算网络拓扑结构，其中每层分别为边缘设备、边缘域、雾域和云域。节点可能会发生崩溃或拜占庭故障，每个容错域最多可容忍一个错误。系统内部使用 Paxos 和 PBFT 协议来处理崩溃故障和拜占庭故障域中的共识问题。

实验工作负载基于微支付应用，模拟了商业环境中的高频资金转账场景。实验在 Amazon EC2 平台上进行，节点分布在多个虚拟机（VM）上。每个高度为 H-1 及以上的域（即服务器域）分配一个独立的虚拟机。所有客户端均运行在同一台虚拟机上。每个虚拟机配备了 8 核 CPU、15 GB 内存，处理器为 3.50 GHz 的 Intel Xeon E5-2666 v3。

此外，实验还将 Saguaro 与两个现有的对比系统进行对比：AHL（基于协调器的共识机制系统）和 SharPer（基于分片的共识机制系统）。通过与这些对比系统的性能对比，评估 Saguaro 在不同实验场景下的优势。

场景：跨域事务

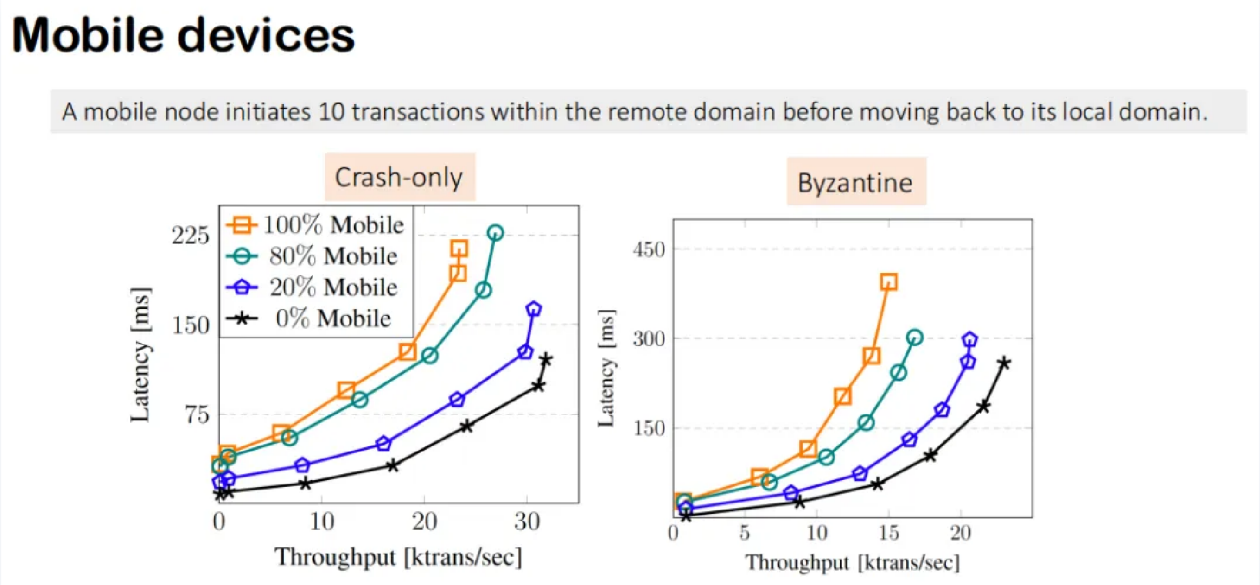




在跨域事务实验中，作者评估了不同跨域事务比例（0%、20%、80%、100%）对 Saguaro 性能的影响。实验在四个 AWS 区域（法兰克福、米兰、伦敦、巴黎）进行，测量了跨域事务的延迟和吞吐量。实验结果表明，当事务中跨域比例较低时，基于乐观的共识协议表现最优，能够以较低的延迟处理更多事务。例如，在 20% 跨域事务的场景下，吞吐量达到 22500 tps，延迟为 105ms。

随着跨域事务比例的增加，基于协调器的共识协议表现出相对于现有系统（如 AHL 和 SharPer）更为优越的性能。特别是在 100% 跨域事务的情况下，Saguaro 的吞吐量比 AHL 高出 63%。即便在存在拜占庭节点的环境下，虽然吞吐量有所下降，Saguaro 依然能够保持较高的性能，这主要得益于其多个协调器域的分布式设计。

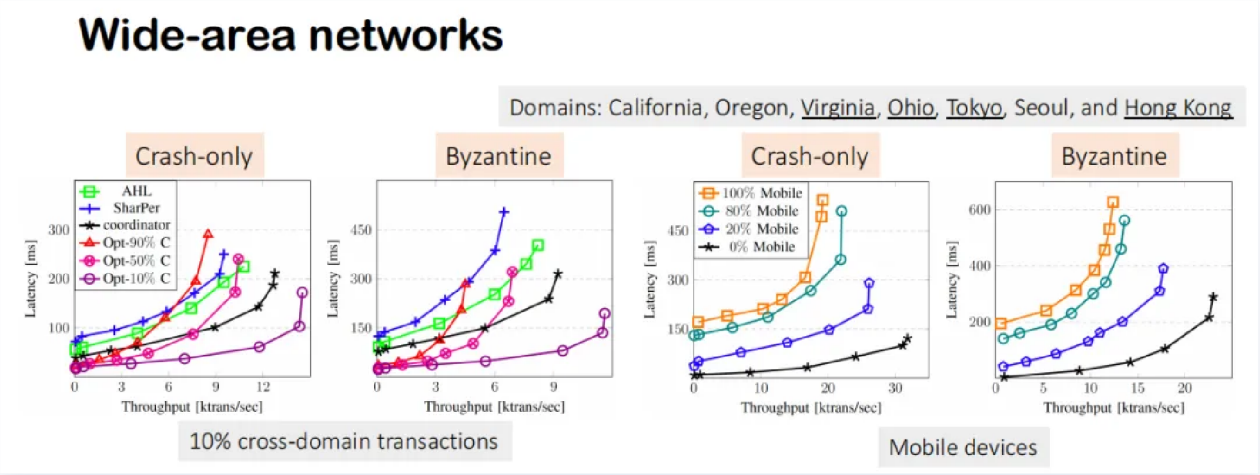
场景：移动事务



在移动事务的实验中，作者评估了不同移动节点比例（0%、20%、80%、100%）对 Saguaro 性能的影响。实验使用了与跨域事务相同的网络环境，并模拟了边缘设备的移动性。实验结果表明，当事务中没有移动节点时，Saguaro 可以处理 31000 tps，且延迟低于 100ms。当增加 20% 的移动节点时，吞吐量仅下降约 4%。而在 80% 和 100% 移动节点的情况下，吞吐量分别为 25700 tps 和 23200 tps，这表明 Saguaro 能够有效处理高比例的移动事务。

然而，在拜占庭节点环境下，由于状态消息的共识成本较高，吞吐量下降了 36%。这些实验结果表明，Saguaro 在支持需要设备移动性的应用（如共享出行）时，能够提供良好的性能和扩展性。

场景：广域网



在广域网络实验中，作者评估了长距离网络对 Saguaro 性能的影响，尤其是在跨域事务和移动事务的场景下。实验将域分布在 7 个不同的 AWS 区域（加利福尼亚、俄勒冈、弗吉尼亚、俄亥俄、东京、首尔和香港）。实验结果表明，在处理低冲突工作负载时，Saguaro 的基于乐观的共识协议仍然表现出色，能够高效处理事务。然而，随着冲突的增加，特别是在高冲突工作负载中，基于乐观的共识协议的性能受到了较大影响。远程域之间的事务处理需要更多时间来解决不一致，导致事务中止的频率增加。

在跨域事务的场景下，基于协调器的共识协议相比 AHL 和 SharPer 展现出更强的优势，特别是在 100% 跨域事务时，吞吐量比 AHL 高出 63%。在移动事务的实验中，尽管网络距离较长导致了更高的延迟，Saguaro 依然表现出了较高的吞吐量，尤其是在处理大量移动节点时。当移动设备比例从 0% 增加到 100% 时，Saguaro 的吞吐量仅减少了 38%，展示了其在广域网环境中的高效性能和良好的扩展性，尤其是在支持需要节点移动的应用场景中。

总结与评价

总结：

总体而言，实验评估结果表明，基于协调器的协议在处理所有类型的工作负载时优于 SharPer 和 AHL，展示了其在广域网环境中的可扩展性，特别适合大规模部署。虽然乐观协议在低冲突工作负载下能够通过避免跨域通信高效处理事务，但在高冲突工作负载中，协议的性能显著下降，主要是由于不同域之间账本的不一致，导致大量依赖数据的事务被中止。尽管 SharPer 在近距离域间的性能较好，AHL 在远距离域间则展现了更优的性能，得益于其基于协调器的共识协议。此外，Saguaro 在广域网环境中也能够高效支持节点的移动性，证明了其在动态环境中的有效性，尤其是在支持需要节点移动的应用场景中。

评价：

- 作者在巧妙利用了边缘计算的分布式特点，将共识协议零成本的融入到了网络中，这一叙事十分出色。
- 本篇文章的实验设置恰到好处，结构简单却目标明确。

- 此篇论文以特定场景设计了三个共识模块，将原本三个独立的内容结合到了一起。

附文

以下是两个实验时使用的对比系统，由于其属于其他论文的部分，这里只进行简单介绍：

SharPer（"Sharding Permissioned Blockchains over Network Clusters"）是 M. J. Amiri 等人在 2021 年提出的一种基于分片的许可区块链系统，旨在通过分布式网络集群提高区块链的性能和可扩展性。SharPer 采用分片技术，将区块链网络划分为多个处理单元，每个分片负责处理特定的事务和数据，从而降低每个节点的负载，提升吞吐量。系统通过高效的跨分片协调机制，确保事务的一致性和安全性，特别是在处理大规模事务时，能显著减少延迟。对于许可区块链，SharPer 还在安全性和访问控制方面提供了优化方案。实验结果表明，SharPer 在集群环境中能够显著提高处理能力，特别适合需要高吞吐量和大规模数据管理的应用场景，如金融、供应链等。

AHL（"Towards Scaling Blockchain Systems via Sharding"）是 H. Dang 等人在 2019 年提出的区块链分片扩展方案，旨在解决传统区块链在大规模应用中遇到的性能瓶颈问题。AHL 采用分片技术，将区块链网络分成多个小的子网络（分片），每个分片独立处理自己的事务和账本，从而实现更高的吞吐量和低延迟。为了保持全局一致性，AHL 提出了跨分片通信和同步机制，确保分片之间的事务和数据能够正确地协作。AHL 的设计特别关注如何高效地管理分片之间的依赖和协调，以避免过多的跨分片交互导致性能下降。实验结果表明，AHL 在高吞吐量和低延迟的要求下，能够显著提高区块链系统的可扩展性，适用于需要快速事务确认的大规模应用场景，如金融支付和供应链管理。