极端规模容器的高效数据推送和一致性保障

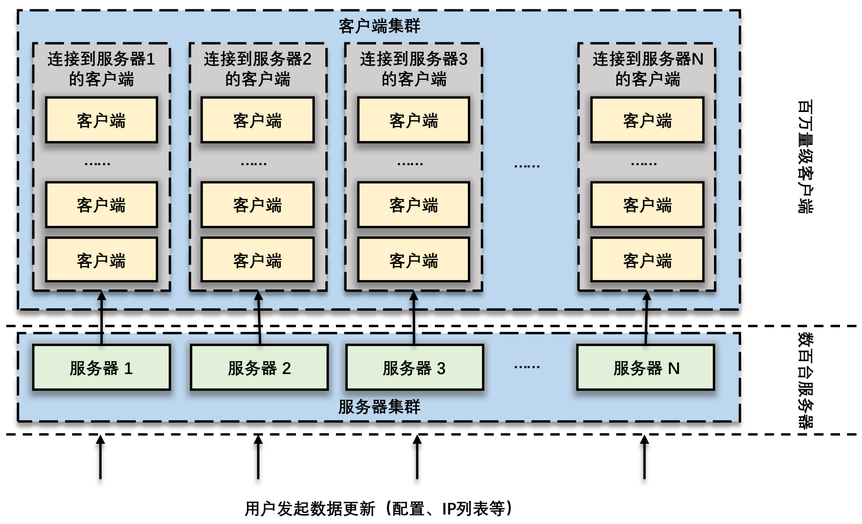
唐震，宋傲，纪树平，王伟
软件工程技术研究开发中心
tangzhen12@iscas.ac.cn wangwei@otcaix.iscas.ac.cn

工作介绍

配置管理在很多现实场景下有重要应用：一个典型场景是云服务提供商的**极端规模**容器实例的运维管理，在此场景中配置更新由数百量级的服务器节点推送至**百万量级的客户端**节点。配置更新的**快速可靠推送**是保障服务正常运行的关键环节之一。

然而，现有的解决方案在**性能**、**扩展性**和**可靠性**方面存在局限性。例如，Raft等保障顺序一致性的共识算法扩展能力有限，且网络分区时节点数量较少的集群不可用；Gossip等保障最终一致性的数据同步算法存在收敛速度慢的问题，难以满足极端规模集群的配置管理需求。

为应对上述挑战，我们提出了一种适用于大规模集群的可扩展、高性能的配置管理解决方案。该解决方案引入了新的**层次化推送机制**及**改进的发布-订阅机制**，并使用客户端节点的计算能力协助进行数据推送，实验结果表明，与SOFAJRaft相比，我们的方法平均推送吞吐率可提升最多5.59倍。



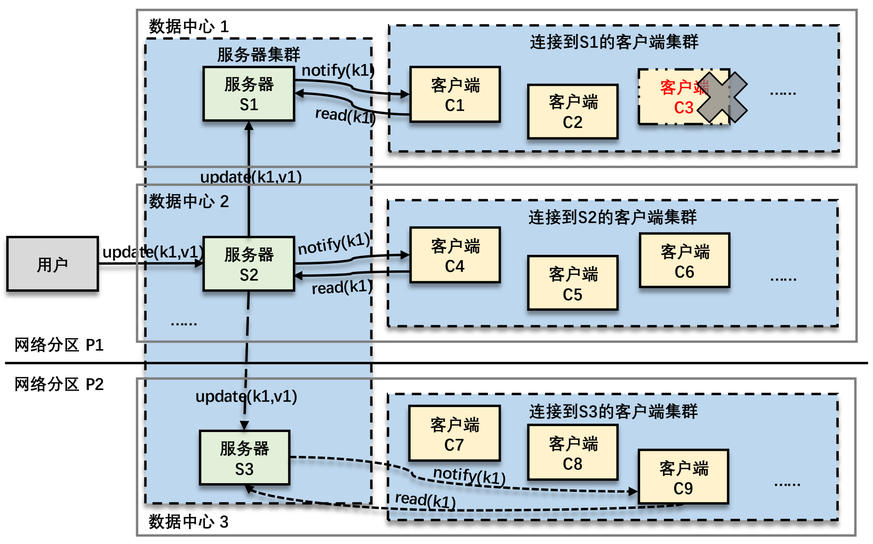
问题分析

在百万量级客户端场景下需要解决的关键问题是如何**高效**、**可靠**地将更新数据发送至对它感兴趣的客户端。除了解决这一性能问题之外，在此规模的集群中还存在如下的新常态带来的新需求：

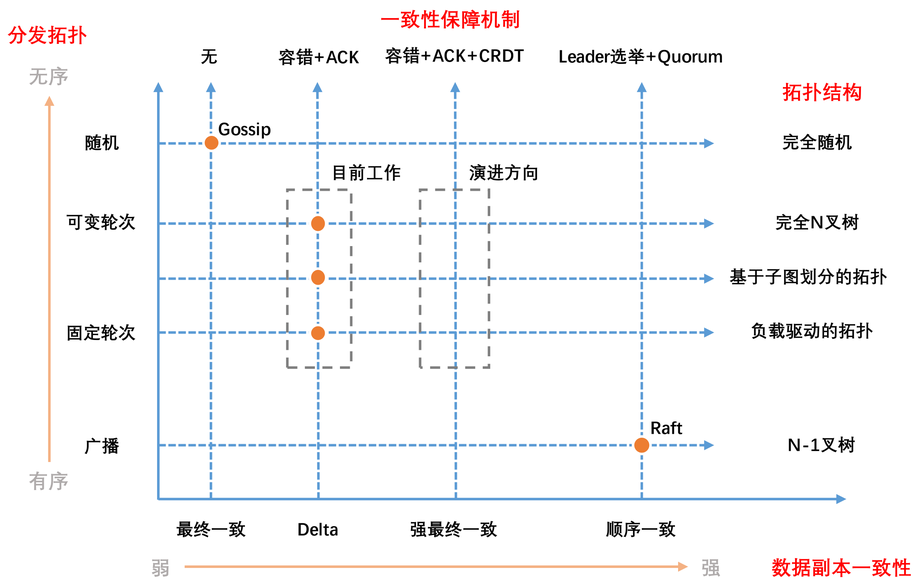
- **节点失效**或**网络不可达**这类小概率事件在大规模集群中变得很常见；这要求解决方案可以应对节点不可达的场景。
- **多地多数据中心**的部署方式已成为承载大规模集群的有效途径，多个数据中心间由于链路故障而导致**网络分区**的情形也时有发生；在此场景下需要保证所有网络分区内的服务仍然可用。

因此，我们的配置管理解决方案需要满足如下需求：

- **数据按时到达**：在规定时间内将配置更新推送至可达网络分区内的可达节点。
- **容错和多分区可用**：可容忍节点失效和网络分区，并保证出现网络分区时所有分区可用。
- **最终一致性**：节点失效恢复或网络分区恢复后最终可收到所有配置更新。



技术细节



完整的配置管理解决方案可以解构为**分发拓扑**和**一致性保障机制**两方面：

分发拓扑：我们提出了基于生成树的多轮次传播并衍生出3种拓扑；

- **基于完全N叉树的拓扑**：将有序节点列表转换为完全N叉树。
- **基于子图划分的拓扑**：将有序的节点列表划分为若干个规模不超过C的小集群，小集群内全连通，小集群间构建无向边。
- **负载驱动的拓扑**：节点检测到负载超过阈值时转发后续的推送任务；推送任务被转发到位于N个等分区域中的共计随机N个节点，这些节点负责将更新广播至所在区域的其他节点。

一致性保障机制：我们基于“**绕开不可达节点**”的容错机制、**心跳检查**机制和**ACK位图**机制保障最终一致性。

- **“绕开不可达节点”**：将失效节点的更新推送任务交由上层节点完成。
- **心跳检查和ACK**：每个节点向邻居节点交换心跳信息；推送更新时源节点维护完整的ACK位图，拓扑中的下层节点维护以它为根的子树的ACK位图。

实验结果

我们基于阿里巴巴开源服务配置中心Nacos实现了原型系统并进行实验验证。实验流程如下：

- 在阿里云ECS主机上部署Nacos Server集群，同时启动大规模的推送客户端，将这些客户端订阅到Nacos Server集群上。
- 通过基于JMeter的负载发生工具PAS，向服务端集群施加不同等级的配置更新请求；服务端在收到配置更新请求后，将这些更新的配置推送给订阅的客户端上。
- 遍历客户端，收集客户端日志。分析客户端日志得到配置更新推送的时间分布。

本实验在至多100个节点组成的Nacos集群中，对比基于完全N叉树的算法（$N=10$）和SOFAJRaft的数据推送延迟。实验使用至多100个节点，每个节点开启3000个客户端连接，以产生至多30万个客户端连接。使用PAS发出恒定TPS=100的请求。

大规模集群下与SOFAJRaft相比，我们的方法平均推送吞吐率可提升最多5.59倍，可以应对更严苛的推送数据收敛时间窗口。

10服务器端，3万客户端

	Tree	SOFAJRaft	吞吐率提升
RT	117.08ms	322.72ms	2.75x
80%RT	117.57ms	361ms	3.07x
95%RT	159.41ms	390ms	2.44x
99%RT	327.92ms	408ms	1.24x

20服务器端，6万客户端

	Tree	SOFAJRaft	吞吐率提升
RT	226.97ms	1047.71ms	4.61x
80%RT	225.17ms	1180ms	5.24x
95%RT	308.18ms	1266ms	4.12x
99%RT	530.33ms	1292ms	2.44x

50服务器端，15万客户端

	Tree	SOFAJRaft	吞吐率提升
RT	534.98ms	2300.02ms	4.29x
80%RT	549.27ms	3060ms	5.59x
95%RT	712.32ms	3395ms	4.76x
99%RT	1075ms	3503ms	3.26x

100服务器端，30万客户端

	Tree	SOFAJRaft	吞吐率提升
RT	1202.67ms	4052.6ms	3.37x
80%RT	1244.47ms	4605ms	3.72x
95%RT	1483.83ms	5049ms	3.40x
99%RT	2025.24ms	5193ms	2.56x