

郭凯,刘杰,方毅,等. 2025. 面向地震预警场景的千万量级预警信息推送服务研究. 中国地震,41(2):339~353.

# 面向地震预警场景的千万量级 预警信息推送服务研究

郭凯<sup>1)</sup> 刘杰<sup>1)</sup> 方毅<sup>2)</sup> 董霖<sup>2)</sup> 刘佳鹏<sup>2)</sup> 侯建民<sup>1)</sup>

1) 中国地震台网中心, 北京 100045

2) 每日互动科技股份有限公司, 杭州 030024

**摘要** 面向地震预警场景中震中区域海量人口及时获得预警信息的减灾需求,解决目前每秒约数十万级别的信息推送能力瓶颈,研究了一种海量预警信息高并发分层级联广播和区域定向推送技术,实现亚秒级千万量级预警信息的推送能力。为验证方法性能,采用95台高性能服务器搭建预警信息多层级联发布平台,其中45台高性能服务器模拟一千万用户进行实验。通过在CM端采用CPU核心绑定以及Zookeeper集群管理等技术,提升单服务器推送性能和集群线性增加的稳定性,在不考虑实际网络延时的情况下具备了秒级千万量级预警信息推送能力,一千万条预警信息整体推送延时在500ms以内,达到了秒级推送一千万条预警信息的性能要求,并通过区域定向推送实际环境检验进一步验证了系统推送能力。

**关键词:** 级联广播 定向广播 边缘计算

[文章编号] 1001-4683(2025)02-0339-15 [中图分类号] P315 [文献标识码] A

## 0 引言

地震预警系统(Earthquake Early Warning System, EEWS)可以在破坏性地震波到达前几秒到几十秒发出预警信息(Allen et al, 2003),是能够有效减轻地震灾害的重要技术手段之一。地震预警系统需要在破坏性地震波到达前,以最短的时间延迟,及时将预警信息推送到地震周边一定范围内的公众和相关企业政府用户(Hoshiba et al, 2008),达到紧急避险和决策应对的目的。相关研究表明,预警时间为3s可使人员伤亡减少14%,预警时间10s可使人员伤亡减少39%,预警时间20s可以减少高达63%的人员伤亡(夏玉胜等, 2000)。目前地震预警系统已经在全球多个国家进入试运行或实际运行,如日本铁路的UREDAS(Urgent Earthquake Detection Alarm System)(Kodera et al, 2021)、墨西哥的SAS(Seismic Alert System)(Suárez et al, 2021)和美国的Elarms(Earthquake Alarm System)(McBride et al, 2020)。预警信息的快速发布是整个地震预警系统在社会服务中的重要体现(蔡寅等, 2019; 申源等, 2024)。国外应急信息的发布主要采用小区广播技术(Axelsson et al, 2007),如美国的商用

[收稿日期] 2024-04-26 [修定日期] 2024-07-03

[项目类别] 地震科技星火项目“面向微信场景的海量预警信息推送关键技术研究”资助

[作者简介] 郭凯,男,1986年生,高级工程师,主要研究方向为人工智能和地震预警。E-mail: guokai@seis.ac.cn

刘杰,通讯作者,男,1965年生,研究员,主要研究方向为地震学。E-mail: liujie@seis.ac.cn

移动警报系统 CMAS (Commercial Mobile Alert System)、无线紧急警报系统 WEA (Wireless Emergency Alerts) (Ngo et al, 2011) 以及日本的地震海啸预警系统 ETWS (Earthquake and Tsunami Warning System) (Musa et al, 2013) 等。目前地震预警发布主要通过预警专用终端、应急广播和短信等方式来实现 (Wang et al, 2021), 其时效性和覆盖范围有限, 尤其是短信的时效性短板非常明显。如美国 Elarms 系统受限于并发能力, 仅推送延时就在 3.5s 以上 (McBride et al, 2023), 而基于目前的手机 APP 信息推送技术, 推送能力在数万每秒至十万每秒 (张盼盼, 2020), 用户基本在数秒甚至分钟级才能收到预警信息, 对于发生在人口密集区域的破坏性地震, 采用这种方式使得预警信息无法及时推送到震中附近人员。

信息推送技术主要分为 PULL 和 PUSH 技术 (Kiziltas et al, 2010), 相关研究表明在推送时效和资源使用效率方面, PUSH 技术更适用于预警信息发布场景 (Minnebo et al, 2014)。随着移动终端的不断发展, 面向移动终端信息推送技术的相关研究和应用也得到了较大发展 (Kweon et al, 2014), 如谷歌的 C2DM (Cloud to Device Messaging) (Doriya et al, 2015) 和苹果推送通知服务 APNS (Apple Push Notification Service) (景洪水等, 2014), 但其实时性和适用范围有限; 很多第三方推送技术也得到了大范围应用, 如极光推送、个推推送等。传统推送方式会将所有用户信息维护在服务端的用户平台, 根据用户在线/离线状态、黑/白名单、地理位置等一系列信息进行信息推送, 集中式的用户管理使得用户筛选和路由寻址成为系统瓶颈 (翟成彤, 2019)。千万级用户量推送可能需要几十秒甚至几分钟, 目前今日头条用户数量 4.5 亿、腾讯新闻用户数量 6 亿, 每个推送平台的推送速率约为 20 万用户/秒 (何凯等, 2016)。在日本“3·11”9.0 级地震中, 已经暴露了存在预警信息发布延时较大的问题 (Fujinawa et al, 2015), 如果用户量进一步提升, 耗时还将线性增长, 因此需要设计一种全新的推送模式和去中心化的分层网络技术架构 (汪科等, 2022), 来解决海量信息高并发定向推送的问题。

本文主要在以下两个方面开展研究:

(1) 研究并实现面向预警场景的信息推送分层级联广播技术架构。按中心节点、区域分中心节点、消息分发节点、设备连接节点等多层结构实现系统分层级联, 各层级采用广播分发方式进行地震预警信息并行发布;

(2) 提升单服务器推送性能和级联推送线性增加服务器的稳定性。通过高速缓存 Redis、消息中间件等技术框架以及 CPU 核心绑定等技术进行性能调优, 实现系统高可用性和动态扩容能力。

## 1 海量信息高并发分层级联广播技术

### 1.1 传统推送技术对比

推送技术的应用对于服务器端的性能是一个极大的挑战, 系统不仅要处理大量用户的并发连接, 还需要处理海量的信息, 负载过高会导致信息下发延时明显增大 (阮伟标, 2017)。传统信息推送架构 (图 1(a)) 是将所有用户信息维护在服务端的用户平台, 查询所有用户以及判断用户信息较为耗时, 千万级用户量可能需要几十秒甚至几分钟。本文研究的分层级联广播并发技术 (图 1(b)) 通过分散连接移动终端设备, 采用边缘计算技术, 利用“去中心化”的思路, 将原先整个用户平台的存储压力和查询压力分摊到每个客户端连接管理器实例

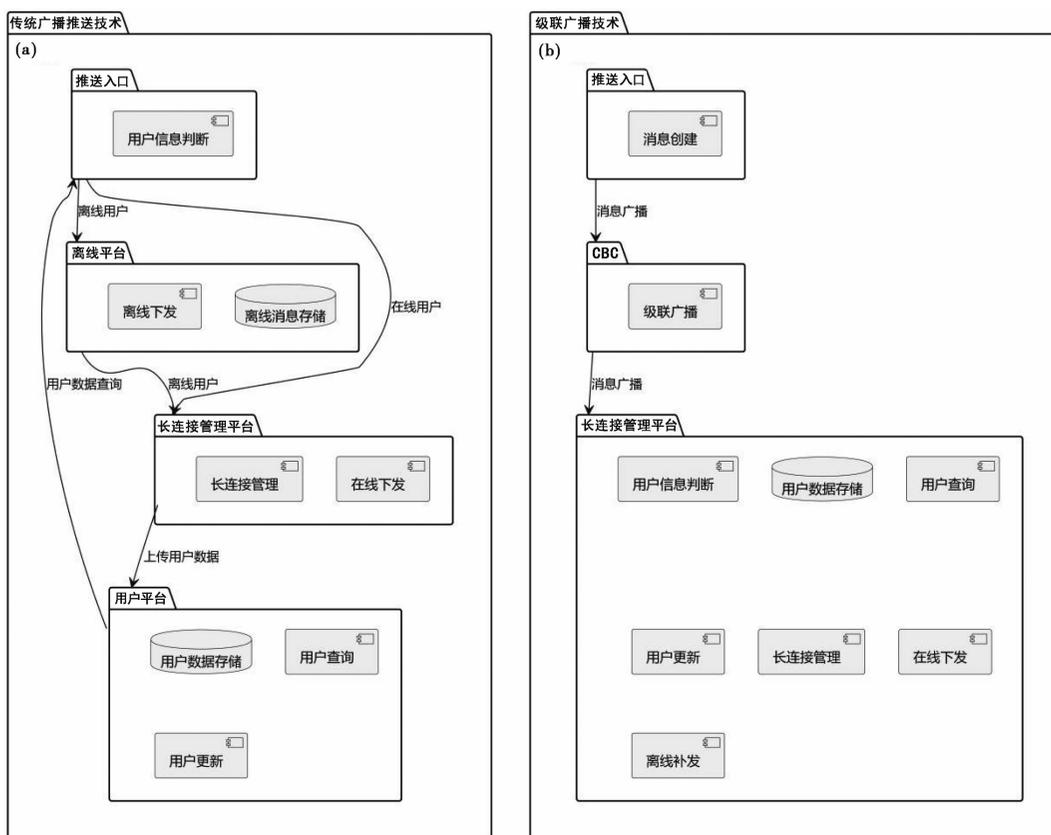


图 1 传统广播推送技术(a)与本文提出的级联广播技术(b)架构对比

(CM)上,可以大大减少查询导致的时间损耗,实现更快的网络服务响应,实现预警消息快速分发,并采用 Zookeeper 进行分布式集群管理,通过单服务器多 CM 部署进行 CPU 核心绑定,实现并发性能优化,提高消息分发的吞吐率。本研究主要使用的符号和描述见表 1。

### 1.2 高并发分层级联广播设计

本文设计的分层级联广播架构主要模块包括:推送管理平台(Push Manager System, PMS)、级联广播中心(Cascade Broadcast Center, CBC)、客户端连接管理器(Connection Manager, CM),整个级联广播架构的主要核心为 CBC 层级模式(图 2)。CBC 模块依赖高速服务框架(High-speed Service Framework, HSF),建立与次级 CBC 之间的远程过程调用框架连接。上级 CBC 在启动时将服务注册到集群后,通过内部 HSF 框架监听暴露端口,等待次级 CBC 连接握手请求,次级 CBC 同样通过 HSF 服务启动注册到集群后,主动向上级 CBC 发送连接握手请求。上级 CBC 在监听过程中接收到消息,之后开始建立与次级 CBC 之间的 HSF(TCP)连接。在次级 CBC 成功连接到上级后,上级 CBC 将会在静态缓存会话表中以次级 CBC 的集群名作为节点标志。

PMS 为用户提供创建地震消息并发布推送的界面入口和 API 入口,如图 3 所示。CBC 采用级联广播技术理念,分为四级级联广播结构,第一层为 cbc-start,负责接收 PMS 的地震预警消息并广播到所有 IDC(Internet Data Center)机房;第二层为 cbc-idc-start,作为当前

表 1 本文所使用的符号和描述

符号	描述
PMS	地震预警推送管理平台实例
CBC	级联广播中心实例
CM	客户端连接管理器实例
SDK	客户端手机软件
$m$	单层 CBC 实例数量
$M_{CBC}$	整体 CBC 实例数量
$X$	每一层与下一层实例数量比
$L$	集群 CBC 层级总数
$T_{CBC}$	CBC 实例间总耗时
$t$	CBC 实例间耗时
$m_{CM}$	CM 总数
$m_{SDK}$	每 CM 实例 SDK 总数
$T_{CM}$	底层 CBC 到 CM 总耗时
$T_{SDK}$	CM 发送到 SDK 总耗时
$T_{ALL}$	整体集群总耗时
$\lambda$	SDK 网络延迟期望值
$T'_{ALL}$	传统集群架构总耗时
$t'$	传统集群业务层到 CM 的总耗时
$\lambda'$	传统集群 SDK 网络延迟期望值

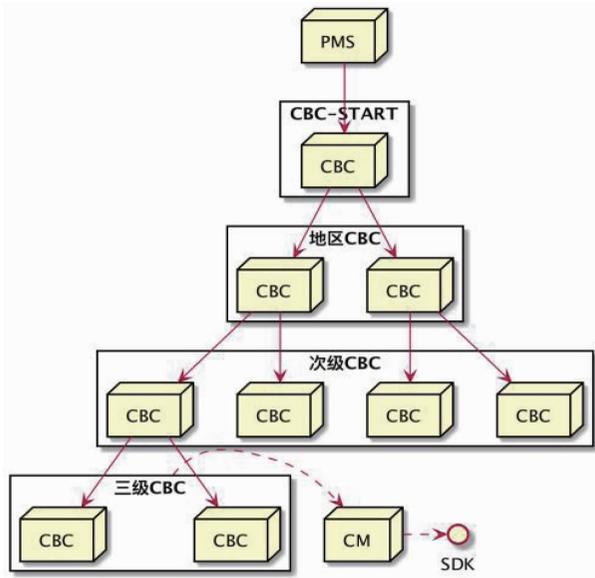


图 2 CBC 层级模式示意图

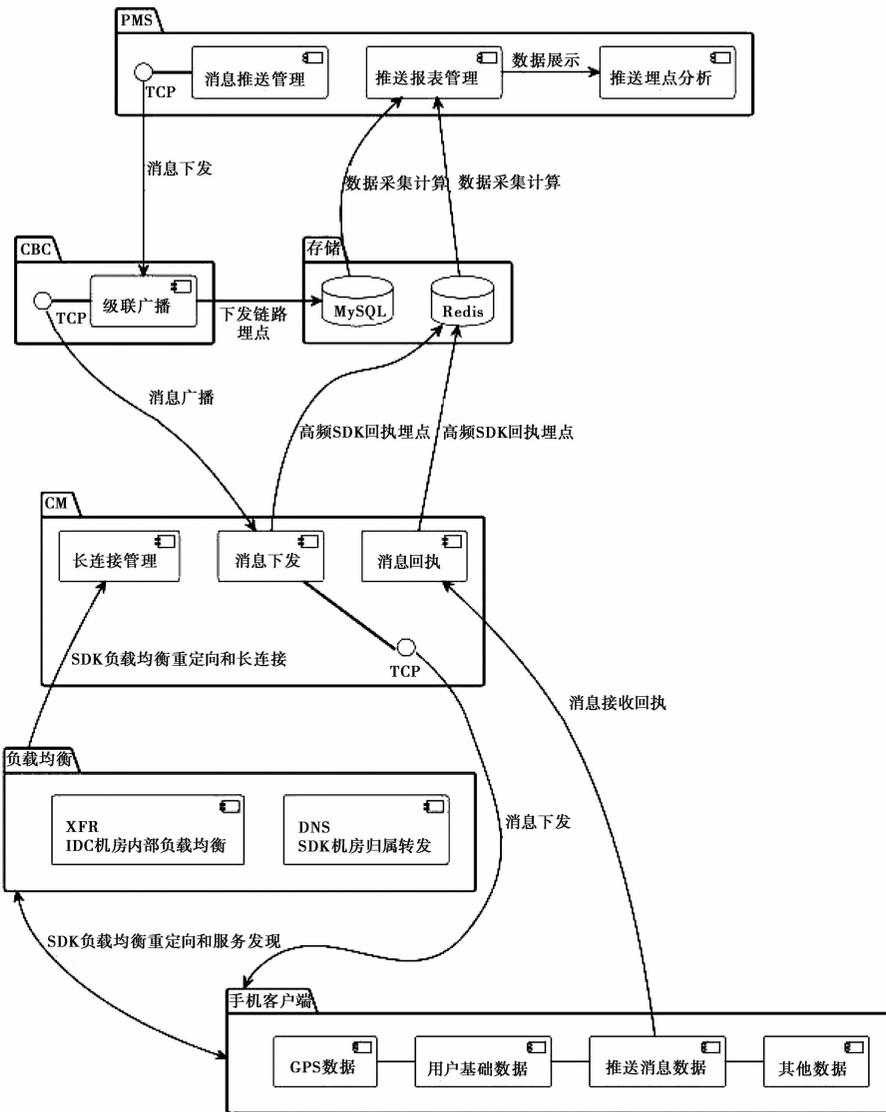


图3 分层级联广播总体架构

IDC 机房机内 CBC 广播中心的入口,接收到 cbc-start 的消息后开始在本机房内开始级联广播;第三层为 cbc-region,作为级联广播的中转节点;第四层为 cbc-region-secondary,收到广播消息后向下发送到 CM。这样就将信息推送由单线程推送扩展到多线程并行推送,极大地满足亚秒级下发的技术要求。由 CM 缓存用户信息,每个 CM 都维护了各自连接用户的信息。具体推送步骤为:

(1) 获取目标消息 (Task) 对应的目标识别码 (TaskID)。

建立目标分发系统,目标分发系统包括:  $L$  个分发层级的广播中心模块和第  $L$  分发层级中广播中心模块对应连接的配置管理模块,其中,  $L$  指目标分发系统分发层级数,第  $i$  分发层级的广播中心模块的数量为  $X_i (i=1,2,\dots,L)$ ,  $X$  指任一分发层级与下一分发层级之间的

目标模块的连接数量,所述目标模块是指可连接到的下一分发层级中广播中心模块或者配置管理模块,并可以表示每个子集 CBC 实例所包含的数量。因此,对于第  $i$  层 CBC 实例,所包含的 CBC 实例数量  $m_{\text{CBC}}$  为  $m_i = X^{i-1}$ ,则整个级联系统中所包含的 CBC 实例总数  $M_{\text{CBC}}$  为

$$M_{\text{CBC}} = \sum_{i=1}^L m_i = X^0 + X^1 + X^2 + \dots + X^{L-1} \quad (1)$$

例如在本文的测试集群中,令  $L = 3, X = 2$ ,则  $M_{\text{CBC}} = 7$ 。

(2) 获取目标时间列表  $T = (T_1, T_2, T_3, \dots, T_i, \dots, T_{L_0})$ ,其中,  $T_i = X_0^i \times t, X_0^i$  指用于构建所述目标分发系统对应的  $X, t$  指所述目标分发系统中任一分发层级向下一分发层级发送消息所需的时间,  $L_0$  指  $X_0$  对应的目标分发系统的层级数。

则对于一个有  $L$  层的集群,总共需要下发的链路为  $L-1$  层,则集群的整体消耗时间为

$$T_{\text{CBC}} = t(L - 1) \quad (2)$$

(3) 为了更好地平衡分发层级数量和整体广播耗时,将目标分发系统分为每层连接的目标模块数量可变的第一单元和每层连接的目标模块数量固定的第二单元。遍历  $T$  且当  $T_i$  大于等于预设的时间阈值  $T_0$  时,将所述目标分发系统划分成第一单元和第二单元;所述第一单元包括第一分发层级至第  $i-1$  分发层级的广播中心模块,其中,任一分发层级中单个广播中心模块与下一分发层级中  $X^m$  个广播中心模块连接 ( $m = 1, 2, \dots, i-1$ );所述第二单元包括第  $i$  分发层级至第  $L$  分发层级的广播中心模块,其中,任一分发层级中单个广播中心模块与下一分发层级中  $X$  个广播中心模块或者配置管理模块连接。例如,在一个 5 层的分发系统中,前 3 层中每个广播中心模块均与下一分发层级中 10 个广播中心模块连接,后 2 层的分发层级中每个广播中心模块均与下一分发层级中 2 个广播中心模块连接,这样一方面减少了整体的层级数量,另一方面又控制了每一层的分发时间。

(4) 基于所述第一单元和第二单元,将所述目标识别码逐级由分发层级发送至客户端 CM。在本系统的级联框架中,CM 与底层 CBC 之间可以为一一对应关系,也可以为多对一对应关系。当采用一一对应方式时,则底层 CM 的数量  $m_{\text{CM}}$  为

$$m_{\text{CM}} = m_L = X^{L-1} \quad (3)$$

(5) 已知在消息下发前,已有 SDK 与 CM 建立 TCP 长连接,且由于负载均衡模块作用,每个 CM 模块维护的长连接数量大致相同,因此可以统一设每台的 SDK 连接数量为  $m_{\text{SDK}}$ ,则 SDK 总数  $M_{\text{SDK}}$  为

$$M_{\text{SDK}} = m_{\text{SDK}} \times m_{\text{CM}} \quad (4)$$

在 TCP 连接场景下,CM 作为服务端需要将 SDK 的长连接通道信息缓存在内存中,并从内存中轮询长连接通道,将连接信息放到线程池中进行批量下发,这里需要结合定向推送技术,按照震中区域距离进行优先推送。假设线程总数为  $n$ ,耗时为  $t_{\text{SDK}}$ ,耗时主要为网络延迟,网络延迟整体服从泊松分布,即

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, \dots, n \quad (5)$$

其中,已知其期望为  $\lambda$ ,即认为网络延迟均值为  $\lambda$ ,CM 实例之间为并发关系,因此宏观上认为整体耗时分摊到所有 CM 实例,因此,整体网络耗时为

$$T_{\text{SDK}} = \frac{m_{\text{SDK}} \times m_{\text{CM}} \times \lambda}{qm_{\text{CM}}} = \frac{\lambda m_{\text{SDK}}}{q} \quad (6)$$

根据以上系统架构,整体的系统耗时可以表示为

$$T_{\text{ALL}} = T_{\text{CBC}} + T_{\text{CM}} + T_{\text{SDK}} = t(L - 1) + t + \frac{\lambda m_{\text{SDK}}}{q} = tL + \frac{\lambda m_{\text{SDK}}}{q} \quad (7)$$

而对于传统架构,不存在级联系统,因此可以认为上述公式中  $L = 1$ , SDK 仅依赖负载均衡分散分布在不同的 CM 集群上,而在区域广播架构中,SDK 会依据 Geohash 序列分布在就近的 CM 集群上,并负载均衡。相对于前者,可以认为传统推送架构的 SDK 网络消耗会比区域广播高。

因此,可以将传统架构的耗时表示为

$$T_{\text{ALL}} = t + \frac{\lambda' m_{\text{SDK}}}{q} (t' \approx tL, \lambda' > \lambda) \quad (8)$$

若令  $T_{\text{ALL}} < T'_{\text{ALL}}$ , 即

$$tL + \frac{\lambda m_{\text{SDK}}}{q} < t' + \frac{\lambda' m_{\text{SDK}}}{q} \quad (9)$$

$$tL - t' < \frac{\lambda' m_{\text{SDK}}}{q} - \frac{\lambda m_{\text{SDK}}}{q} \quad (10)$$

$$\frac{q(tL - t')}{m_{\text{SDK}}} < \lambda' - \lambda \quad (11)$$

由于本文目标以千万用户进行广播,因此显然可得  $\frac{q}{m_{\text{SDK}}}$  要远小于 1, 而上述前提满足  $t' \approx tL, \lambda' > \lambda$ , 因此不等式成立。即认为,当区域广播的网络不成为整体系统负担时,在千万用户的场景下级联广播比传统广播模式更具优势。

## 2 区域定向优先广播技术

区域内用户消息推送技术称为基于移动位置服务 (Location Based Services, LBS) (Biswas et al, 2022), 可通过手机型号定位、物联网物理围栏等技术实现。该技术的运用场景与地震预警区域定向推送的区别在于: ① LBS 技术一般用于固定的地理围栏 (Geo-fencing), 继而等待用户进入围栏触发场景用例, 而地震预警区域定向不会在推送前得知围栏范围, 因此两者在用户地理信息的维护和区域搜索算法上有较大区别; ② 在地理围栏技术中, 一般围栏均是多边形, 由于区域复杂, 为了加快算法, 更多作用于小范围的商圈判定, 而地震预警区域一般很大, 使用类似搜索方法会增加大量耗时。

这里借鉴了 LBS 技术中运用到的 GeoHash 技术 (图 4), GeoHash 编码具有如下三个特点: ① GeoHash 用一个字符串来表示经度和纬度两个坐标, 在数据库中可以实现在一列上应用索引, 降低开发维护的成本与难度; ② GeoHash 表示一个矩形区域, 可以合理利用用户的地址编码, 既能明确用户的位置, 又不至于暴露其精确坐标, 有助于用户的隐私保护; ③ 编码的前缀可以表示更大的区域, 字符串前部相似的部分表示距离相近, 从而可以设计相应的位置匹配算法。

GeoHash 将地球表面按经纬度划分为相等大小的多个区域, 将各区域进行编号。由图 4 可知, 区域划分越细, 编号越长。具体计算方式为: 根据划分位数  $n$  得到横排长度  $x$  和纵排

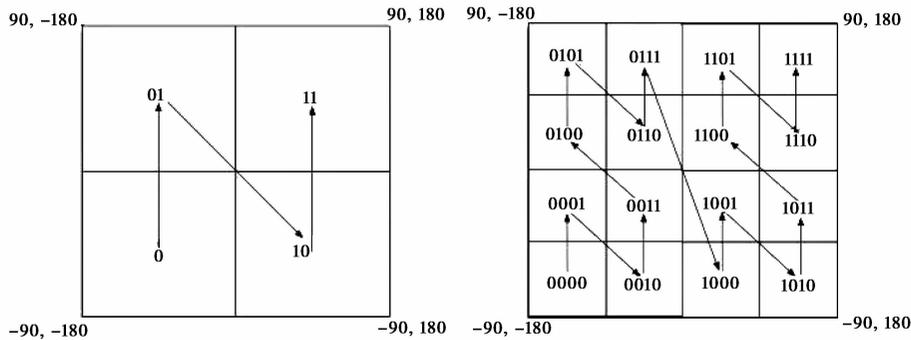


图 4 GeoHash 区域划分编码

长度  $y$ , 以此计算经纬度位于哪一格, 并推导出该格的序号, 即

$$\text{GeoHash} = \text{Hex}(x_i) + \text{Hex}(y_i) \tag{12}$$

$$x_i = \text{latitude} \times \frac{x}{180^\circ} \tag{13}$$

$$y_i = \text{longitude} \times \frac{y}{360^\circ} \tag{14}$$

具体推送步骤为:

(1) 获取目标地理区域对应的网格列表  $B = (B_1, B_2, B_3, \dots, B_q \dots, B_p)$ , 其中,  $B_q$  指第  $q$  个网格,  $q = 1, 2, \dots, p$ ;  $p$  为网格数量;

(2) 获取客户端位置列表  $A = (A_1, A_2, A_3, \dots, A_r \dots, A_m)$ , 其中,  $A_r$  指第  $r$  个客户端 ID 对应的坐标值,  $m$  为客户端 ID 的总数量;

(3) 遍历  $A$  且当  $A_r$  处于  $B_q$  内, 将  $A_r$  对应的客户端 ID 插入至  $B_q$  对应的多个目标客户端列表内, 其中, 任一目标客户端列表  $C = (C_1, C_2, C_3, \dots, C_k \dots, C_{Y0})$ ,  $C_k$  指第  $k$  个目标客户端 ID,  $Y0$  指单个配置管理模块连接的客户端 ID 数量;

(4) 获取目标消息的目标坐标值, 将所述目标消息对应的目标识别码发送至目标区域对应的客户端, 具体包括如下步骤: 当所述目标坐标值处于  $B_q$  内时, 基于  $B_q$  和  $B_q$  相邻的网格构建目标区域, 从所述目标区域中每个网格对应的数据库中提取若干个  $C$ , 将所述目标识别码以逐个目标分发系统的分发层级发送至每个  $C$  中的目标客户端 ID。

### 3 实验验证

#### 3.1 单点性能测试

通过单台服务器部署单 CM 实例的性能测试评估性能瓶颈。由于测试环境从 CBC 到 CM 端, 没有外部网络延时的影响, 为评估其影响, 收集了推送的预警信息网络延时, 发现绝大部分都在 500ms 以内, 但由于其数量较少, 本研究还对每日互动公司从 CM 端推送到实际用户的 170 万条数据的延时进行了统计分析(图 5), 由图可见 90% 的消息耗时在 200ms 内, 96.76% 的消息耗时在 500ms 内, 该时延范围与相关网络时延统计基本一致(杨振辉等, 2018)。因此, 为了实现秒级预警信息下发, 测试环境下从 PMS 到 CM 的耗时需要控制在 500ms 以内, 这样便可为信息向外推送留出足够的时间。具体测试步骤为:

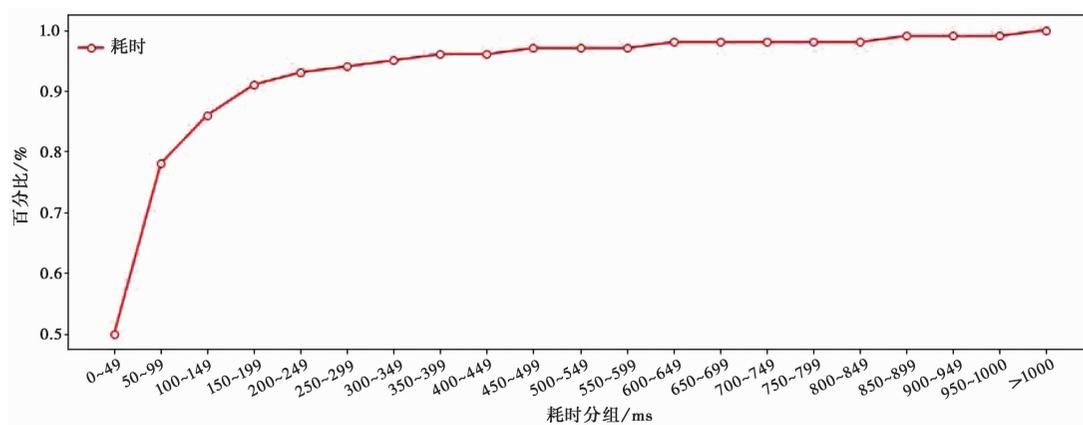


图5 每日互动公司推送的170万条数据时延统计

(1) 在单机器环境下,部署一个CM实例,从10000连接数开始,按每次5000连接逐渐提高连接数,测试CM在500ms下的峰值性能。

(2) 基于测试步骤(1)中得到峰值连接数,在单机环境下,以同样的连接数扩展多台CM,测试单机环境下且不影响整体性能的情况下,最多能同时部署多少CM实例。

(3) 基于测试步骤(2)中得到的CM实例部署情况,以同样的CM实例数和连接数,扩展多台机器,测试多机器部署是否会对整体性能产生影响。

### 3.2 级联性能测试

随着下发信息量的增加,单台服务器单CM实例的延时整体呈上升趋势(图6(a))。测试数据取10次实验的平均值,在500ms以内能够支撑的最大连接数约为6万,但部分实验的耗时超过500ms,因此综合考虑多次实验结果,选择5万作为单个CM实例可接受时延下的最大连接数。当单个CM实例下发5万条消息时,需要占用4~6个CPU核心,故在32核机器上最多可部署5个CM实例(图6(b))。据此,以5万连接作为单个CM实例的峰值性能,在单台机器上逐渐将CM实例数量扩展至5个(每个实例绑定6个CPU核心)。可以看到,当每台机器部署不超过5个CM实例时,耗时几乎没有变化;同时将CM实例部署的机器横向扩展后,整体耗时依然无明显变化(图6(c))。

基于上述测试结果可以得到以下结论:

(1) 单CM实例的峰值认为是50000连接数。

(2) 在32核CPU上,每台机器上可部署的CM实例数为5个(即每个CM实例约占6个CPU核,其中CPU性能会对部署情况产生一定影响),因此基于本文部署服务器的单机峰值性能为500ms发送250000个客户端。

(3) 通过CPU核绑定和本文的级联广播推送架构,横向扩展机器不会对整体耗时产生影响。因此,单机器500ms内下发250000个客户端可以扩展到40台同样配置的机器,达到预警信息同时下发千万量级用户的性能,且耗时控制在500ms以内。

### 3.3 千万量级推送性能测试

基于上述性能测试结果,设计并搭建了千万量级预警信息发布平台,在阿里云新建95台虚拟机,部署硬件清单如表2所示。其中一台机器用作消息下发,部署PMS和CBC

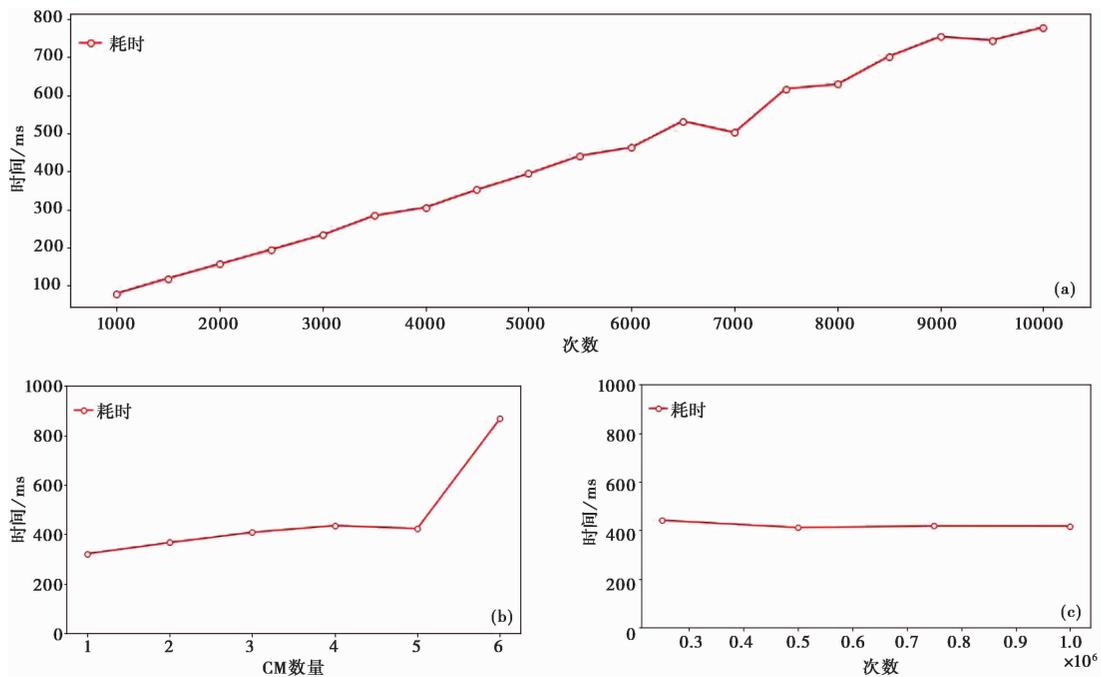


图 6 单机单 CM(a)、单机多 CM(b)和多机多 CM(c)的信息下发时延测试结果

表 2 部署环境硬件清单

序号	名称	操作系统	配置	数量	说明
1	ECS 云服务器	CentOS 8.0 64 bit	CPU32 核,内存 60GB,硬盘 20GB	1	消息下发
2	ECS 云服务器	CentOS 8.0 64 bit	CPU32 核,内存 60GB,硬盘 20GB	3	注册中心
3	ECS 云服务器	CentOS 8.0 64 bit	CPU32 核,内存 60GB,硬盘 20GB	1	提供数据库服务
4	ECS 云服务器	CentOS 8.0 64 bit	CPU32 核,内存 60GB,硬盘 20GB	45	作为 CM 实例,维护长链接
5	ECS 云服务器	CentOS 8.0 64 bit	CPU32 核,内存 60GB,硬盘 20GB	45	模拟客户端

(Start) 模块,三台机器作为注册中心,部署 Zookeeper,利用其树状节点管理中心功能,以节点的各个状态对应服务的健康度。除了服务端微服务以外,将 Redis、Hbase、分布式锁、主从关系等注册到 Zookeeper 上,统一管理,使微服务架构可以方便地发现和调用此类服务,并实现负载均衡。一台机器作为数据库,部署 MySQL 和 Redis,CM 端部署 45 台服务器,每台服务器部署 5 个 CM,并在 8 台服务器上混合布设 CBC (Region) 模块。

为实现千万量级模拟客户端,每台服务器采用主网卡和 4 个虚拟子网,每个 IP 建立 4.5 万个连接,即每台客户端机器发起 22.5 万个长连接,通过共计 45 台机器来发起 1000 万个长连接,每个 CM 实例和模拟连接实例之间进行直连。

整个推送系统模块之间的网络拓扑使用了级联广播的模式,上下层模块维护稳定 TCP 长连接,地震预警消息以消息编号的形式在系统中快速向下广播,直到最底层通信模块,随后将消息下发至模拟链接器。消息从 PMS 下发命令和消息,发送到入口 CBC 中,入口 CBC 采用同步调用的形式,从会话表中提取次级 CBC 的通道信息,通过 HSF 调用到区域 CBC 的

Callback 方法,进行远程回调,从而将消息下发到区域 CBC。区域 CBC 采用异步调用的形式,以同样的 HSF 方法回调到次级 CBC 并下发消息。以此顺序逐层下发,最后由最终层 CBC 进行异步回调 CM,将消息传递到 CM,进而推送到 SDK (图 7)。

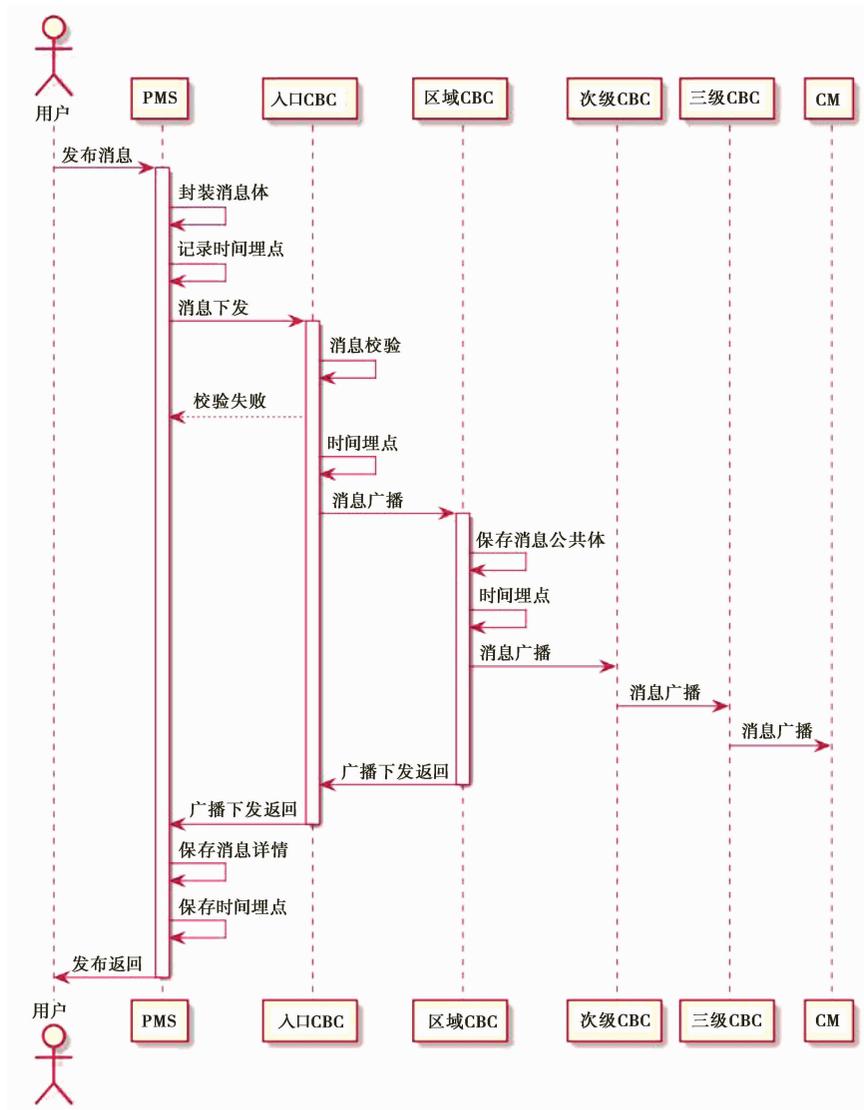


图 7 地震预警信息推送时序图

如图 7 所示,在消息下发功能接收到下发命令和消息后,通过 ProtoBuf (Google Protocol Buffers) 技术解析消息内容,获取到消息中的 APP ID 和 Task ID,根据两条 ID 信息从个推服务器中获取消息公共体内容。同样,将消息内容通过 ProtoBuf 技术反序列化成真消息内容,再采用信息压缩技术转化成二进制流,开始进入发送流程。下发流程开始时,会先遍历该台 CM 模块中维护的 Grizzly 连接会话表,对会话表中缓存的连接判断 APP ID 是否为需要下发的 APP 以及客户端是否在线校验和客户端的地域过滤。为了对系统千万量级性能进

行实际评估,设计测试为:预警信息由 PMS 下发,发给 CBC (Start),广播给 8 个 CBC (Region),再广播给各自连接的 CM,CM 再发给模拟连接。通过 PMS 模块发送预警信息,以每 30s 间隔循环发送 20 次,下发总量 10080006 个长连接,具体测试信息推送情况见图 8。

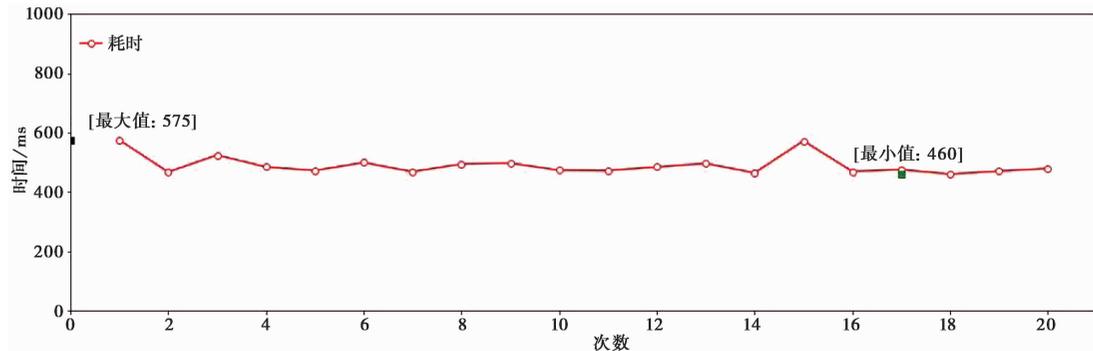


图 8 千万量级预警信息推送系统 20 次耗时

由图 8 可以看到,系统在推送千万量级预警信息时,整体延时在 460~575ms 之间分布,平均延时在 475ms,延时波动较小,表明本文研究的方法具备千万量级预警信推送能力,可以达到亚秒级。相对于传统推送方法,千万量级信息推送往往在数十秒到分钟级,本文设计的级联推送架构体现出海量信息场景下的高时效性能和高并发处理能力。

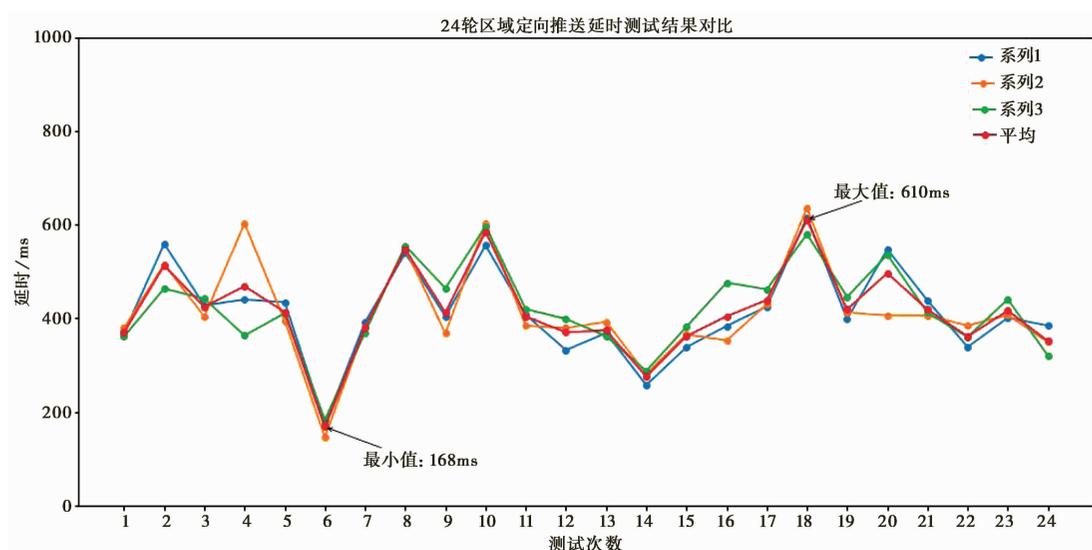
### 3.4 区域定向推送

为了在真实环境中对系统的区域定向推送能力和延时情况进行分析,在天津、四川、山东、福建等地选择了约 100 部手机进行测试。推送方案按照天津、四川、山东、福建 4 个地区模拟 4 个震例信息,每个震例信息分别做 3 次全量广播和 3 次区域定向广播(共计 24 条消息),推送内容和方案如表 3 所示。

表 3 区域定向推送方案

地震编号	震中位置	经纬度	震中参考地名	震级	烈度	广播方式
EQ1	四川凉山	102.98° E, 26.41° N	【测试】四川省凉山州会东县	2.5	II	全量广播
EQ2	四川凉山	102.98° E, 26.41° N	【测试】四川省凉山州会东县	2.5	II	500km 区域定向广播
EQ3	山东泰安	117.08° E, 36.20° N	【测试】山东省泰安市	2.5	II	全量广播
EQ4	山东泰安	117.08° E, 36.20° N	【测试】山东省泰安市	2.5	II	500km 区域定向广播
EQ5	福建泉州	118.67° E, 24.87° N	【测试】福建省泉州市	2.5	II	全量广播
EQ6	福建泉州	118.67° E, 24.87° N	【测试】福建省泉州市	2.5	II	700km 区域定向广播
EQ7	天津武清	117.04° E, 39.38° N	【测试】天津市武清区	2.5	II	全量广播
EQ8	天津武清	117.04° E, 39.38° N	【测试】天津市武清区	2.5	II	500km 区域定向广播

测试数据采用每次客户端收到消息并回执的时间减去服务端消息发出的时间,24 组测试平均延时结果如图 9 所示,可以看到 24 轮测试整体延时均控制在 1s 内,平均延时控制在 500ms 以内。



注: 系列1~系列3代表重复推送的3条地震消息延时。

图9 区域定向推送和全域推送延时情况

## 4 结语

本文采用区域定向优先广播技术和海量用户高并发广播技术,设计了一个海量用户亚秒级地震预警信息发布系统。通过对CM的性能优化,最终提供了一套由PMS 1台、CM 45台、中间件4台(共50台服务器)组成的具备秒级推送千万用户的地震预警信息发布平台。测试表明,该系统在500ms内(不计服务器到终端的网络传输延迟)即可向一千万个并发长连接用户推送地震预警消息,在实际区域定向推送中延时也控制在1s内。该系统的部署和实际应用可以实现震中区域海量用户预警信息的亚秒级推送,提升预警信息服务“最后一公里”的覆盖效能。

相较于传统的推送技术,本文解决的千万量级预警信息推送技术瓶颈主要体现在两个方面:①降低级联系统端到端的网络延时,减少级联系统的业务逻辑,将大量用户管理和计算下沉到CM端,利用广播拓扑结构优化整体的下发耗时链路,以降低系统整体耗时,并在CM系统进行性能优化,使CM具备承载短时间高并发的能力;②结合预警信息发布场景的区域定向推送技术,使海量预警信息可以在亚秒级实现高并发的区域推送。本文研究成果将为我国地震预警系统建设和地震预警信息的广泛应用提供有力的技术支持,可有效减轻未来破坏性地震的人员伤亡和经济损失。此外,该系统也可以用于其他紧急系统的信息发布,具有广阔的应用前景。

## 参考文献

- 蔡寅,张明,赵瑞,等. 2019. 地震预警信息快速发布系统研究. 震灾防御技术, **14**(1): 247~258.
- Fujinawa Y, Noda Y. 2015. 日本地震预警系统在2011年3月11日地震中的效能、不足与改进. 国际地震动态, (7): 4~22.
- 何凯,杜瑞林,魏贵春,等. 2016. 地震预警信息发布时延分析. 大地测量与地球动力学, **36**(11): 1031~1034.

- 景洪水,冉萌. 2014. iPhone 推送通知服务与防火墙协同优化探讨. 邮电设计技术, (9):6~9.
- 阮伟标. 2017. 分布式消息推送系统的设计与实现. 硕士学位论文. 广州: 华南理工大学.
- 申源,郭凯,梁厚朗,等. 2024. 国家紧急地震信息服务系统的设计、开发及实践. 中国地震, **40**(1):1~23.
- 汪科,陈家慧,吴幼龙. 2022. 基于分层网络的中心化和去中心化编码缓存方案. 中国科学院大学学报, **39**(2):224~231.
- 夏玉胜,杨丽萍. 2000. 地震预警(报)系统及减灾效益研究. 西北地震学报, **22**(4):452~457.
- 杨振辉,柯腾辉,周壮,等. 2018. 基于 TCP 的移动网络时延优化研究与实践. 电信技术, (10):9~12,19.
- 翟成形. 2019. 基于长连接的分布式消息推送系统设计与实现. 硕士学位论文. 西安: 西安石油大学.
- 张盼盼. 2020. 高性能地震预警信息分发系统研究与开发. 现代信息科技, **4**(19):116~120,125.
- Allen R M, Kanamori H. 2003. The potential for earthquake early warning in southern California. *Science*, **300**(5620):786~789.
- Axelsson K, Novak C. 2007. Support for cell broadcast as global emergency alert system. Norrköping: Institutionen för Teknik och Naturvetenskap.
- Biswas P, Dandapat S K, Sairam A S. 2022. Ripple: an approach to locate  $k$  nearest neighbours for location-based services. *Inf Syst*, **105**:101933.
- Doriya R, Suraj K, Chakraborty P. 2015. A C2DM based framework for detection and notification of terrorist activities on android platform. *Lect Notes Software Eng*, **3**(4):325~329.
- Hoshihara M, Kamigaichi O, Saito M, et al. 2008. Earthquake early warning starts nationwide in Japan. *Eos Trans Am Geophys Union*, **89**(8):73~74.
- Kiziltas S, Akinci B, Gonzalez C. 2010. Comparison of experienced and novice cost estimator behaviors in information pull and push methods. *Can J Civ Eng*, **37**(2):290~301.
- Kodera Y, Hayashimoto N, Tamaribuchi K, et al. 2021. Developments of the nationwide earthquake early warning system in Japan after the 2011  $M_w$  9.0 Tohoku-Oki earthquake. *Front Earth Sci*, **9**:726045.
- Kweon K, Park J, Lee J, et al. 2014. Energy efficient IP reachability for push service in NATted LTE systems. In: *IEEE International Conference on Communications*. Sydney: IEEE, 2526~2531.
- McBride S K, Bostrom A, Sutton J, et al. 2020. Developing post-alert messaging for ShakeAlert, the earthquake early warning system for the West Coast of the United States of America. *Int J Disaster Risk Reduct*, **50**:101713.
- McBride S K, Sumy D F, Llenos A L, et al. 2023. Latency and geofence testing of wireless emergency alerts intended for the ShakeAlert® earthquake early warning system for the West Coast of the United States of America. *Saf Sci*, **157**:105898.
- Minnebo W, Van Houdt B. 2014. A fair comparison of pull and push strategies in large distributed networks. *IEEE/ACM Trans Netw*, **22**(3):996~1006.
- Musa A, Kuba H, Kamoshida O. 2013. Earthquake and tsunami warning system for natural disaster prevention. In: *Resch M M, Wang X, Bez W, et al. Sustained Simulation Performance 2012*. Berlin, Heidelberg: Springer, 81~91.
- Ngo P, Wijesekera D. 2011. Enhancing the usability of the commercial mobile alert system. In: *5th IFIP WG 11.10 International Conference on Critical Infrastructure Protection V*. Hanover: Springer, 137~149.
- Suárez G, Espinosa-Aranda J M, Cuéllar A, et al. 2021. Evaluation of the seismic alert system of Mexico (SASMEX) during the June 23, 2020, Oaxaca earthquake ( $M_w$  7.4). *Nat Hazards*, **108**(3):3085~3098.
- Wang Q P, Zheng C, Xiao J, et al. 2021. Overview of earthquake early warning information release system in Fujian Province, China. *IOP Conf Ser: Earth Environ Sci*, **632**(2):022033.

## Research on Ten-million-level Early Warning Information Push Service for Earthquake Early Warning Scenarios

Guo Kai<sup>1)</sup>, Liu Jie<sup>1)</sup>, Fang Yi<sup>2)</sup>, Dong Lin<sup>2)</sup>, Liu Jiapeng<sup>2)</sup>, Hou Jianmin<sup>1)</sup>

1) China Earthquake Networks Center Center, Beijing 100045, China

2) Daily Interactive Technology Co., Ltd., Hangzhou 030024, China

**Abstract** In earthquake early warning scenarios, the timely dissemination of warning information to the large population in the epicentral region is crucial for mitigating disaster impacts. However, current systems face limitations in pushing hundreds of thousands of messages per second. To address this challenge, we have developed a high-concurrency, hierarchical cascade broadcast, and region-specific push technology capable of delivering tens of millions of early warning messages per second. To validate this technology, we constructed a multi-level cascade release platform comprising 95 high-performance servers, with 45 of these servers simulating 10 million users for experimental purposes. We implemented CPU core binding and Zookeeper cluster management techniques on the Central Management (CM) side to enhance both the individual server push performance and the stability of linear scalability within the cluster. Excluding actual network latency, the system demonstrated the ability to push tens of millions of early warning messages per second. The overall push delay for 10 million messages was maintained within 500 milliseconds, thereby meeting the performance requirement of pushing 10 million messages per second. Additionally, the system's push capability was further validated through practical environmental testing, confirming its region-specific push functionality.

**Keywords:** Cascade broadcasting; Targeted broadcasting; Edge computing