

SACC 2014中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2014

发现架构之美

易到消息推送平台 设计与实现

余庆

2014-09-15

消息推送平台基本要求

- 快速可靠送达
- APP和平台一直保持长连接
- 平台需要支持高并发连接数

消息推送平台现状

- 比较成熟，百家争鸣
- 第三方平台包括：百度推送、个推、极光推送等等

需要推送的消息类型

- 系统级消息：实时性要求不高
 - 通知消息
 - 推广消息
- 应用级消息：实时性要求较高
 - 派单消息
 - 聊天消息

为什么要自建消息推送平台

- 我们主要推送应用级消息
- 送达率和实时性要求高
- 第三方消息推送平台无法满足需求

原有消息推送平台

- 基于ejabberd的消息下发：消息较臃肿
- 基于mongodb轮询，简单但低效
- 原有实现无法做到线性扩容
- 送达率约99%

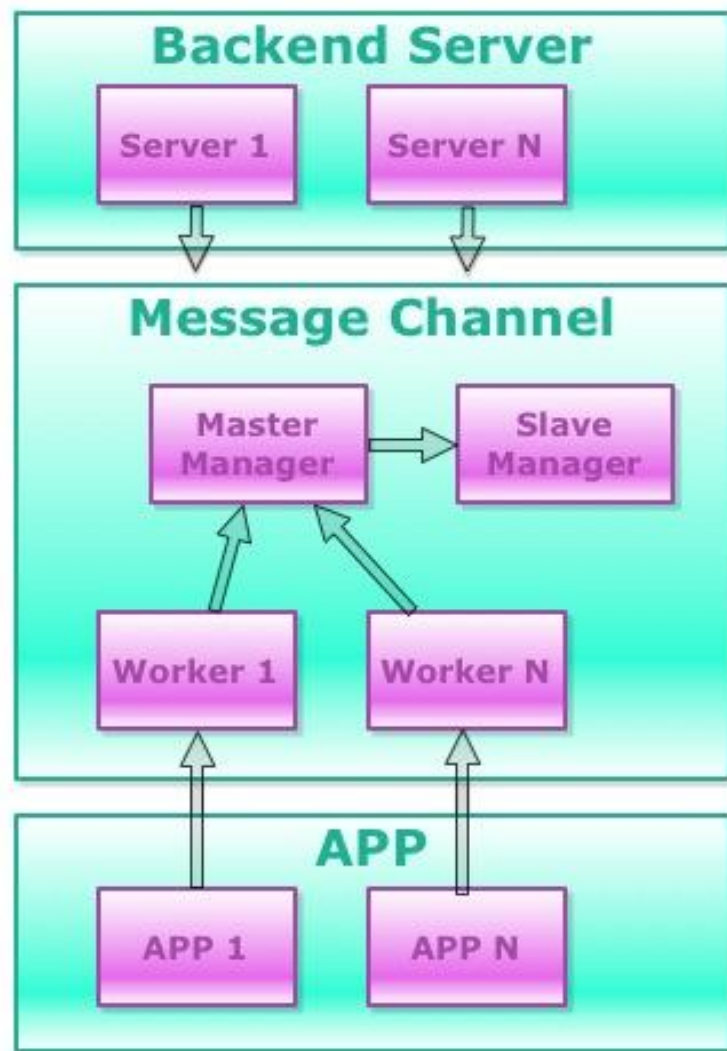
新消息推送平台设计目标

- 支持双向消息推送
 - 消息下发
 - 数据上传
- 无单点，支持线性扩容
- 实现消息不落地推送
- 系统支持的用户数过亿
- 单机支持的并发用户数10w+

只是一个消息推送平台？

- 消息推送平台下个阶段——消息通道（MC）
- 支持消息双向推送
- 连接APP和后端的接入层

易到MC架构



架构关键点

- 一个APP只会分配到一台worker上
- APP在线的情况下，必须将消息发送到其对应的worker

架构特点——manager

- manager维护设备到worker的映射关系
- 对连接到worker的APP进行负载均衡
- 采用主从热备方式：只有两台server
- master主动连接slave，同步映射关系数据
- 主失效，从自动接替
- 主恢复后，重新成为主

架构特点——worker

- 消息中转
- 主动连接manager，发送心跳信息
- 对等结构，无限扩展

APP连接MC过程

- APP请求manager分配worker
- APP直接连接该worker，通过ping保持长连接

后台发送消息过程

- 后台请求manager分配worker
- 后台直接连接该worker发送消息

MC RPC



MC RPC优势

- 实现统一的接入层，大大降低service进程数
- 天然长连接，避免建连开销
- 天然session机制，避免对每个请求都要进行用户身份认证
- service跨语言：可以用C、PHP等语言编写

MC实现

- C语言实现
- 二进制私有协议
- 网络通信采用libfastcommon
 - 直接封装epoll、kqueue和port
 - 网络超时采用时间轮
- 消息队列 + 事件驱动

消息发送机制

- 消息队列缓存，有最大长度限制
- 生产者——消费者模型
- 发送时不能跨线程，采用事件方式通知对应的工作线程
- 每轮epoll处理结束后，根据消息事件推送消息

manager fail over机制

- 我们不需要第三方组件！
- manager server采用主从结构
- 两台server采用心跳方式探测对方是否存活
- APP和后端先连接master，失败后再连接slave
- slave接替master，后来master恢复后，slave会踢掉已经建立的连接

Q&A

THANKS

SequeMedia
盛拓传媒

IT168.com
www.it168.com

ChinaUnix

ITPUB