GOTC

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE , OPEN WORLD

「音视频性能优化」专场

开源视频服务器，
凭什么SRS能做到全球Top1？

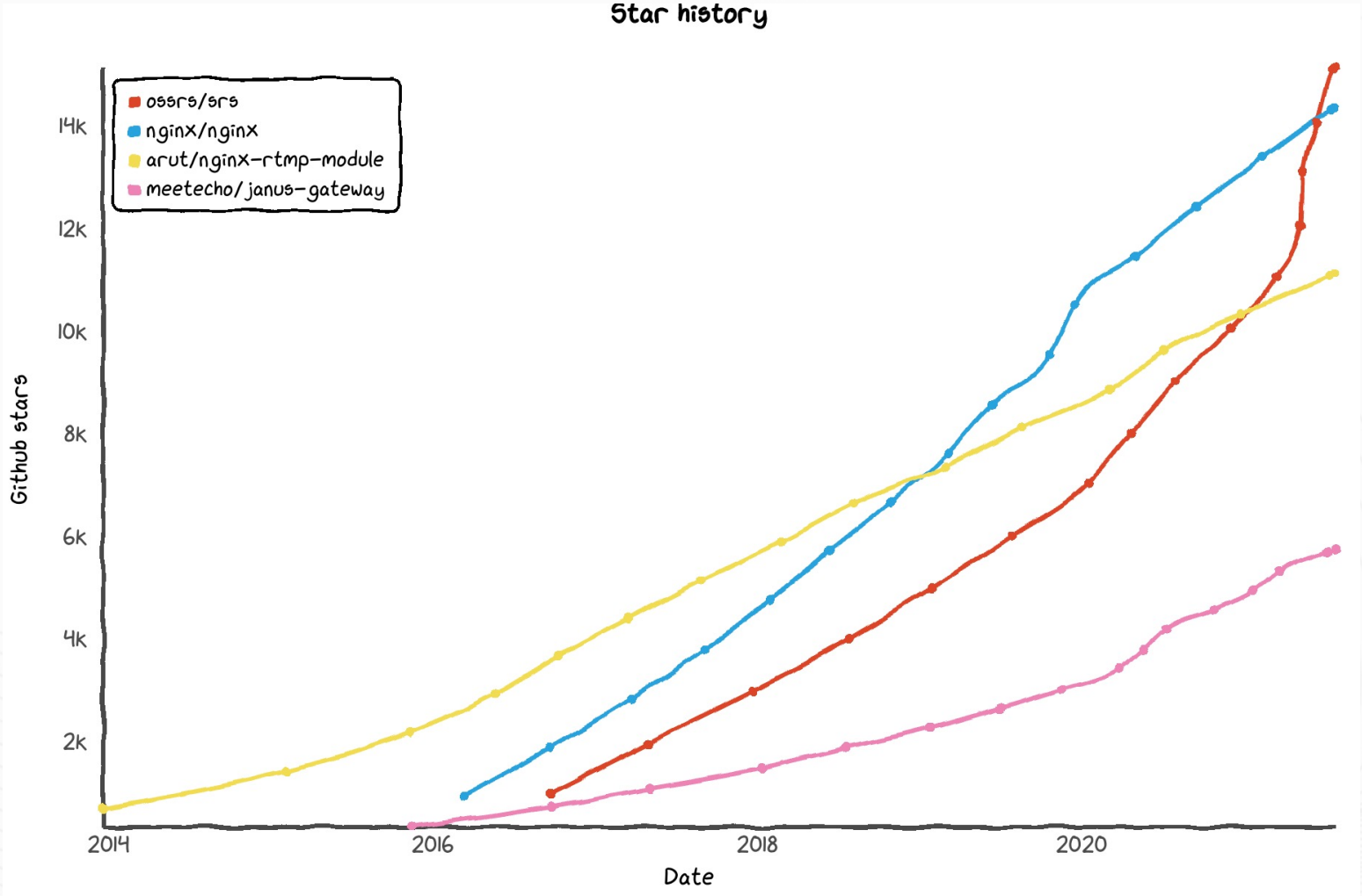
杨成立 2021.07

目录

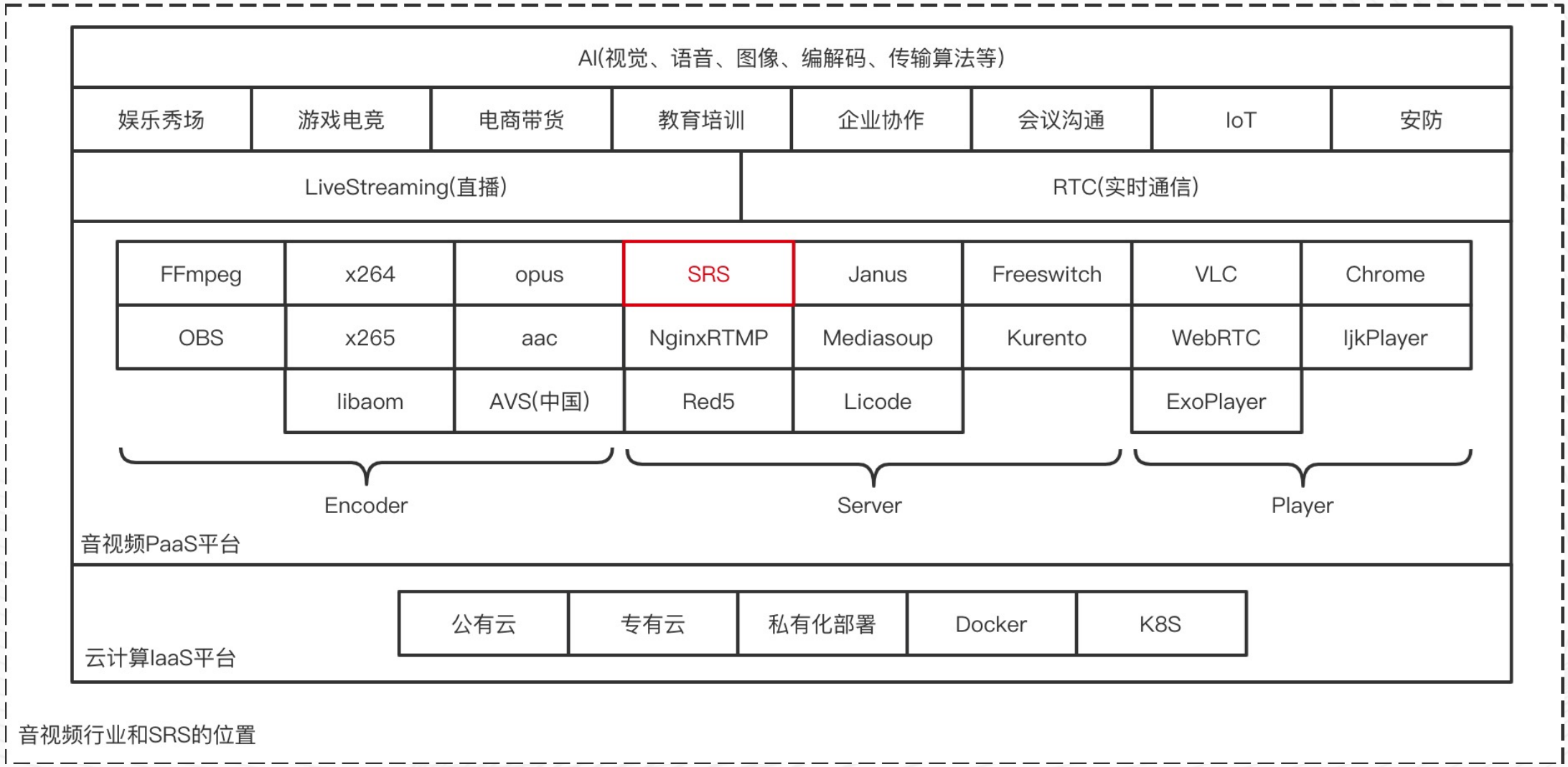
SRS开源服务器

微信扫描二维码，关注我的公众号

全球Top1开源视频服务器



国内音视频的业务爆发



音视频行业和SRS的位置

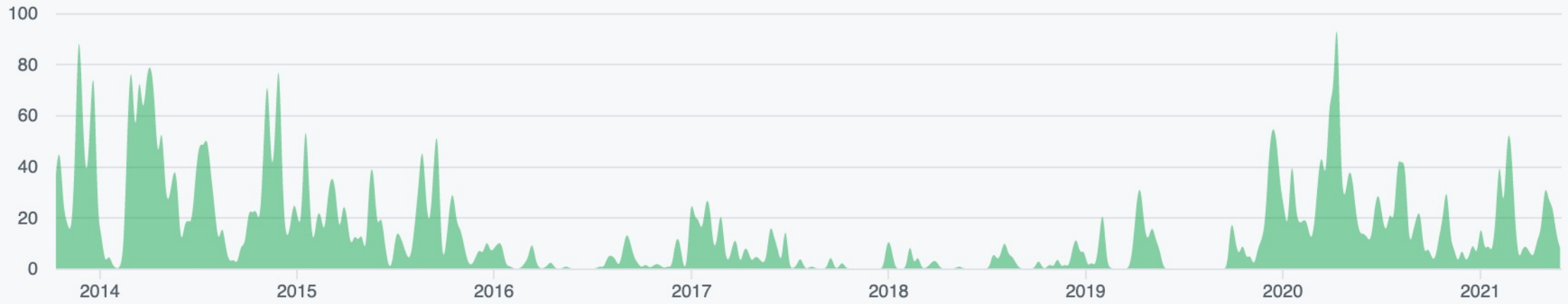
持续不断更新和完善

GOTC

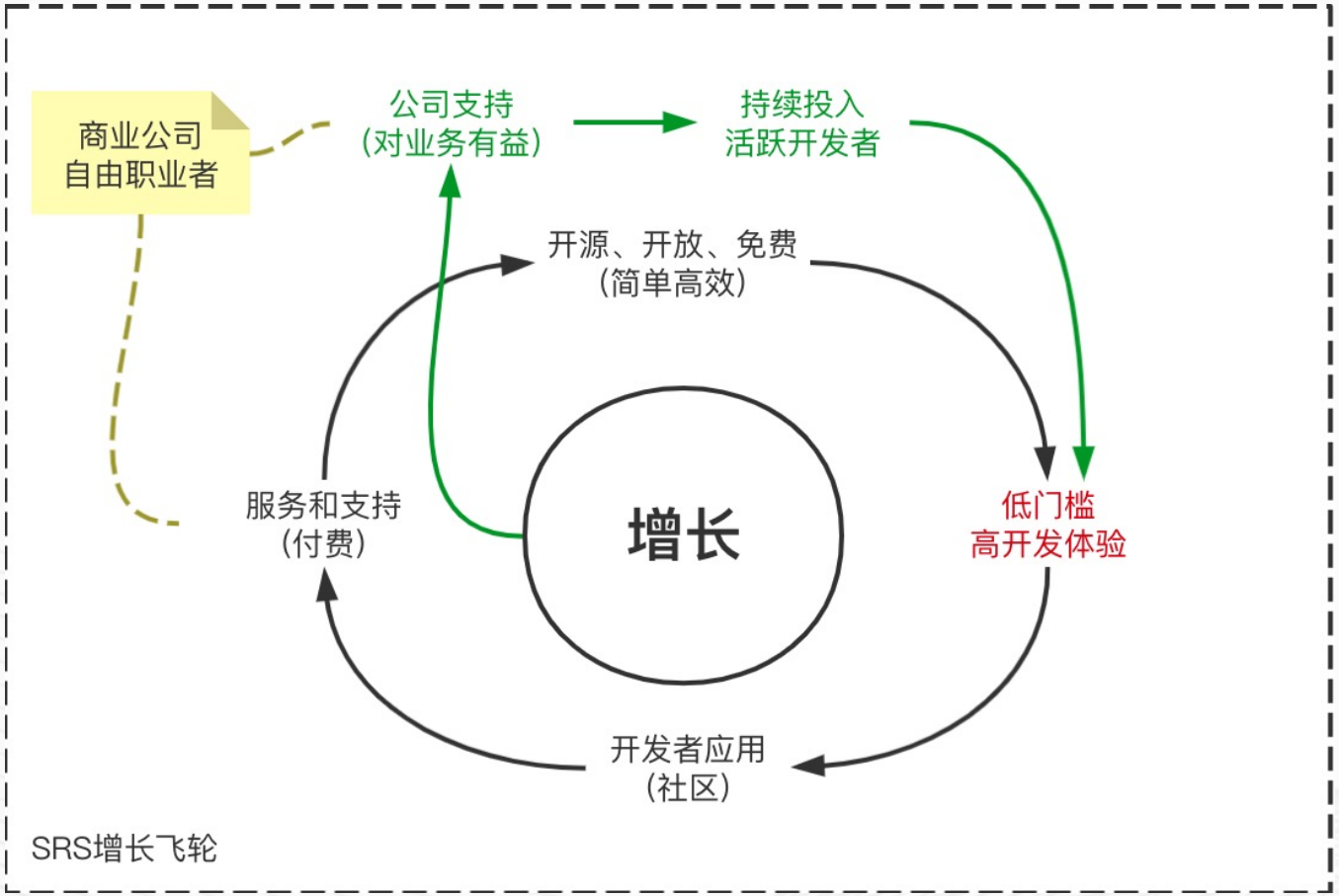
Oct 13, 2013 – Jun 2, 2021

Contributions: Commits

Contributions to 4.0release, excluding merge commits and bot accounts



成长飞轮：开源、应用、服务



- SRS(Core)开放免费：MIT授权，不做直接商业化
- SRS付费服务：培训、咨询、运维、支持、开发、托管
- 行业核心痛点是门槛高，开发者体验差

关键技术解析

- 性能是同类项目的三倍
- “零”依赖编译，全Docker镜像，全链路Demo
- 中英文文档，视频和培训，咨询和支持服务(付费)
- 直播集群，RTC级联(开发中)
- IDE：srs-toolkit-idea(开发中)

关键技术解析

性能是同类项目的三倍

SFU	Clients	CPU	Memory	线程	VM
SRS	4000 players	~94% x1	419MB	1	G5 8CPU
NginxRTMP	2400 players	~92% x1	173MB	1	G5 8CPU
SRS	2300 publishers	~89% x1	1.1GB	1	G5 8CPU
NginxRTMP	1300 publishers	~84% x1	198MB	1	G5 8CPU

SFU	Clients	CPU	Memory	线程	VM
SRS	1000 players	~90% x1	180MB	1	G5 2CPU
Janus	700 players	~93% x2	430MB	24	G5 2CPU
SRS	950 publishers	~92% x1	132MB	1	G5 2CPU
Janus	350 publishers	~93% x2	405MB	23	G5 2CPU

> Note: CentOS7, 600Kbps, ECS/G5-2.5GHZ(SkyLake), SRS/v4.0.105, NginxRTMP/v1.2.1。虽然系统有8CPU但只能使用单个CPU，选择8CPU是因为只有8CPU的内网带宽才能到10Gbps。

关键技术解析

GOTC

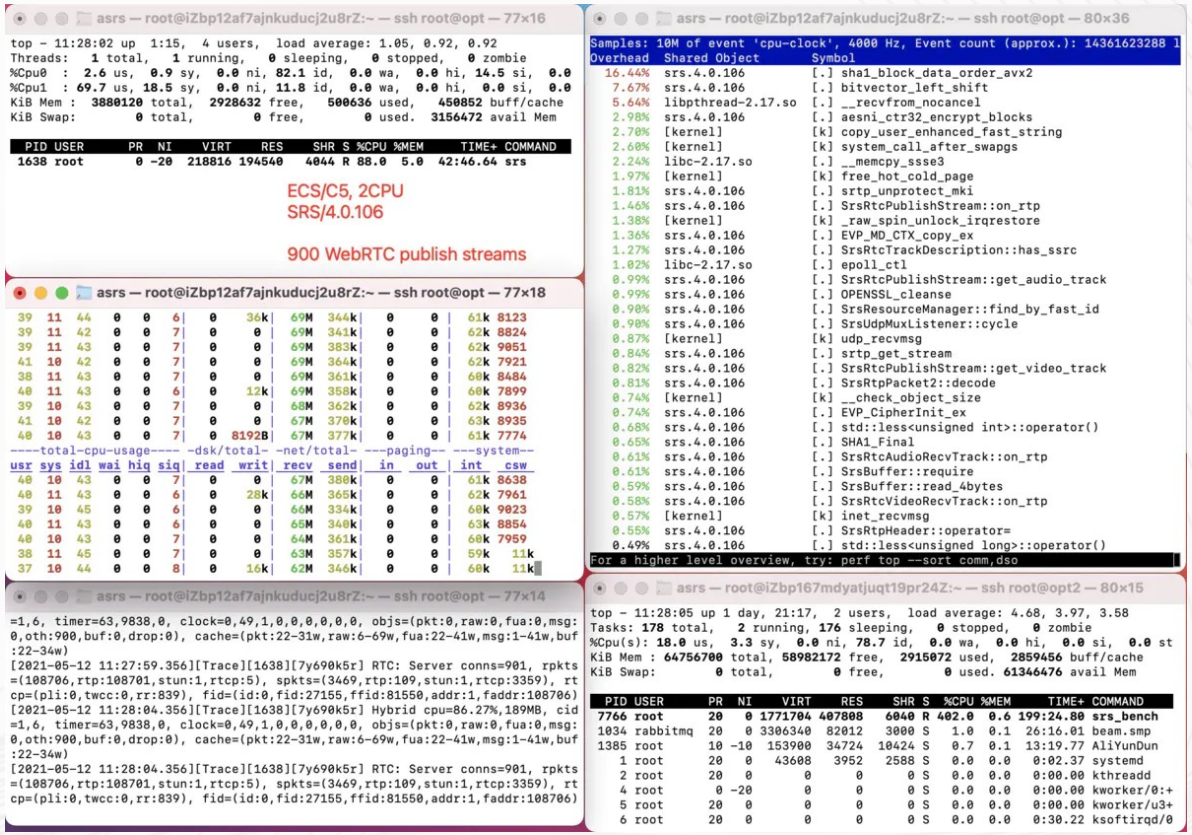
服务器并发架构

- 第一代高并发架构，多线程架构。
  - 一般无法解决C10K问题，线程之间的同步和竞争
  - Adobe AMS，Apache HTTP Server，Janus WebRTC Server
- 第二代高并发架构，单线程架构。
  - C10K问题得到比较好的解决，存在异步回调问题，解法：协程
  - Nginx，SRS 4.0(协程)，MediaSoup
- 第三代高并发架构，隔离的多线程架构。
  - 反向代理Envoy线程模型，比Nginx更高性能
  - Envoy和Nginx都是事件驱动，但是Envoy是完全非阻塞。
  - SRS 5.0将使用隔离的多线程+协程架构。

关键技术解析

性能和Benchmark

- Benchmark是性能分析和提升的基础。
- srs-bench支持RTMP、WebRTC压测。
- srs-bench支持回归测试。



关键技术解析

直播性能优化

- 内存交换性能，直播性能提升3倍。

```
srs_error_t SrsRtmpConn::do_playing(SrsLiveSource* source, SrsLiveConsumer* consumer, SrsQueue
{
    mw_msgs = _srs_config->get_mw_msgs(req->vhost, realtime);
    mw_sleep = _srs_config->get_mw_sleep(req->vhost);

    while (true) {
        consumer->wait(mw_msgs, mw_sleep);

        if ((err = consumer->dump_packets(&msgs, count)) != srs_success) {
            return srs_error_wrap(err, "rtmp: consumer dump packets");
        }

        if (count > 0 && (err = rtmp->send_and_free_messages(msgs.msgs, count, info->res->stre
            return srs_error_wrap(err, "rtmp: send %d messages", count);
        }
```

关键技术解析

GOTC

RTC性能优化

- 端口复用需要频繁查找，用vector/unordered_set代替map
- **性能热点 = 函数执行效率 x 函数执行次数**，无拷贝NACK减少函数执行次数
- 多线程：写日志，防止block

```
srs_error_t SrsRtcRecvTrack::on_nack(SrsRtpPacket** ppkt)
{
    rtp_queue_->set(seq, pkt);
    *ppkt = NULL;
```

关键技术解析

GOTC

探索中的性能优化

- UDP协议栈：sendmmsg
- UDP协议栈：GSO
- 内存拷贝：ZERO_COPY
- 多线程：写录制文件和HLS，隔离的线程
- 硬件加速：指令集，DPDK等。

关键技术解析

GOTC

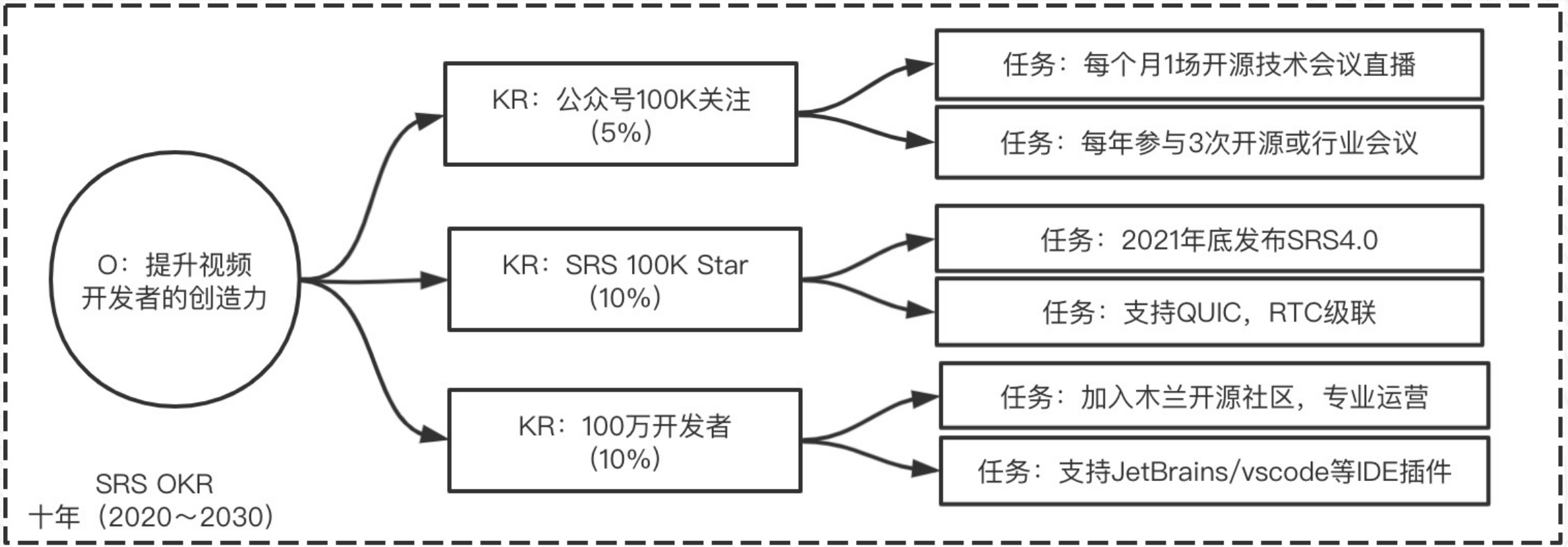
高精度定时器

- epoll服务器有定时器误差问题。
- 繁忙的UDP服务器更明显，EAGAIN概率小。

```
nfd = epoll_wait(fds, 3ms);
for (int i = 0; i < nfd; i++) {
    int active_fd = fds[i];
    // 如果有上千的fd需要读写，那么处理完可能不止3ms，比如20ms。
 }
 // 处理完fd，检查定时器一定超时，而且比预期的多17ms了，那么看起来这个
 // 定时器就是37ms才唤醒，而不是20ms唤醒。
```

十毫秒定时器	<10ms	<15ms	<25ms	<30ms	<35ms	<40ms	<45ms
改进前	0	18	7	8	2	2	1
改进后	0	46	3	0	0	0	0

OKR：提升开发者创造力



GOTC

THANKS

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE