

# 对 PHP 未来发展的一些思考

韩天峰

PHPConChina 历年完整 PPT 下载：

<https://github.com/ThinkDevelopers/PHPConChina>

视频回看地址：

<https://www.itdks.com/Home/Act/apply?id=5366>

PPT 版权归属 PHPCon 组委会和嘉宾本人所有，请勿通过其他渠道提供下载

# PHPConChina 官方渠道

- 官网: <http://www.phpconchina.com/?o=ppt>
- 公众号: PHPCon
- 纪念品购买: <https://k.weidian.com/H3=4lVho>
- 客服咨询: PHPConChina (个人微信号)
- 官方QQ群: 34449228 (加群注明 PHPCon)



扫码关注了解行业最新动态

# 关于我

- 学而思网校首席架构师，基础架构部负责人
- Swoole 开源项目创始人、核心开发者
- 10 年 PHP & C/C++ 程序员



# QUIC & HTTP3



```
$ ping www.taobao.com
PING www.taobao.com (61.155.221.227): 56 data bytes
64 bytes from 61.155.221.227: icmp_seq=0 ttl=57 time=41.575 ms
64 bytes from 61.155.221.227: icmp_seq=1 ttl=57 time=26.661 ms
64 bytes from 61.155.221.227: icmp_seq=2 ttl=57 time=32.011 ms
64 bytes from 61.155.221.227: icmp_seq=3 ttl=57 time=19.793 ms
64 bytes from 61.155.221.227: icmp_seq=4 ttl=57 time=28.245 ms
64 bytes from 61.155.221.227: icmp_seq=5 ttl=57 time=35.931 ms
64 bytes from 61.155.221.227: icmp_seq=6 ttl=57 time=20.394 ms
64 bytes from 61.155.221.227: icmp_seq=7 ttl=57 time=14.487 ms
64 bytes from 61.155.221.227: icmp_seq=8 ttl=57 time=18.146 ms
64 bytes from 61.155.221.227: icmp_seq=9 ttl=57 time=48.171 ms
64 bytes from 61.155.221.227: icmp_seq=10 ttl=57 time=9.687 ms
64 bytes from 61.155.221.227: icmp_seq=11 ttl=57 time=17.870 ms
64 bytes from 61.155.221.227: icmp_seq=12 ttl=57 time=14.779 ms
64 bytes from 61.155.221.227: icmp_seq=13 ttl=57 time=56.601 ms
64 bytes from 61.155.221.227: icmp_seq=14 ttl=57 time=12.896 ms
64 bytes from 61.155.221.227: icmp_seq=15 ttl=57 time=26.382 ms
64 bytes from 61.155.221.227: icmp_seq=16 ttl=57 time=17.469 ms
64 bytes from 61.155.221.227: icmp_seq=17 ttl=57 time=62.159 ms
64 bytes from 61.155.221.227: icmp_seq=18 ttl=57 time=27.583 ms
64 bytes from 61.155.221.227: icmp_seq=19 ttl=57 time=28.528 ms
64 bytes from 61.155.221.227: icmp_seq=20 ttl=57 time=44.908 ms
64 bytes from 61.155.221.227: icmp_seq=21 ttl=57 time=46.884 ms
64 bytes from 61.155.221.227: icmp_seq=22 ttl=57 time=14.566 ms
64 bytes from 61.155.221.227: icmp_seq=23 ttl=57 time=43.202 ms
64 bytes from 61.155.221.227: icmp_seq=24 ttl=57 time=38.174 ms
64 bytes from 61.155.221.227: icmp_seq=25 ttl=57 time=29.035 ms
64 bytes from 61.155.221.227: icmp_seq=26 ttl=57 time=35.059 ms
64 bytes from 61.155.221.227: icmp_seq=27 ttl=57 time=44.881 ms
^C
--- www.taobao.com.danuoyi.tbcache.com ping statistics ---
28 packets transmitted, 28 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 9.687/30.574/62.159/13.790 ms
```

```
$ ping github.com
PING github.com (192.30.255.112): 56 data bytes
64 bytes from 192.30.255.112: icmp_seq=0 ttl=45 time=280.471 ms
64 bytes from 192.30.255.112: icmp_seq=1 ttl=45 time=300.097 ms
64 bytes from 192.30.255.112: icmp_seq=2 ttl=45 time=281.626 ms
64 bytes from 192.30.255.112: icmp_seq=3 ttl=45 time=280.407 ms
64 bytes from 192.30.255.112: icmp_seq=4 ttl=45 time=287.906 ms
64 bytes from 192.30.255.112: icmp_seq=5 ttl=45 time=276.205 ms
64 bytes from 192.30.255.112: icmp_seq=6 ttl=45 time=334.626 ms
64 bytes from 192.30.255.112: icmp_seq=7 ttl=45 time=275.754 ms
64 bytes from 192.30.255.112: icmp_seq=8 ttl=45 time=310.584 ms
64 bytes from 192.30.255.112: icmp_seq=9 ttl=45 time=293.072 ms
64 bytes from 192.30.255.112: icmp_seq=10 ttl=45 time=279.954 ms
64 bytes from 192.30.255.112: icmp_seq=11 ttl=45 time=405.666 ms
64 bytes from 192.30.255.112: icmp_seq=12 ttl=45 time=323.586 ms
64 bytes from 192.30.255.112: icmp_seq=13 ttl=45 time=306.074 ms
64 bytes from 192.30.255.112: icmp_seq=14 ttl=45 time=338.786 ms
64 bytes from 192.30.255.112: icmp_seq=15 ttl=45 time=302.687 ms
64 bytes from 192.30.255.112: icmp_seq=16 ttl=45 time=482.425 ms
Request timeout for icmp_seq 17
64 bytes from 192.30.255.112: icmp_seq=18 ttl=45 time=291.431 ms
64 bytes from 192.30.255.112: icmp_seq=19 ttl=45 time=286.769 ms
Request timeout for icmp_seq 20
64 bytes from 192.30.255.112: icmp_seq=21 ttl=45 time=305.004 ms
64 bytes from 192.30.255.112: icmp_seq=22 ttl=45 time=297.843 ms
64 bytes from 192.30.255.112: icmp_seq=23 ttl=45 time=347.862 ms
64 bytes from 192.30.255.112: icmp_seq=24 ttl=45 time=300.904 ms
Request timeout for icmp_seq 25
64 bytes from 192.30.255.112: icmp_seq=26 ttl=45 time=359.685 ms
64 bytes from 192.30.255.112: icmp_seq=27 ttl=45 time=391.588 ms
64 bytes from 192.30.255.112: icmp_seq=28 ttl=45 time=280.913 ms
64 bytes from 192.30.255.112: icmp_seq=29 ttl=45 time=281.725 ms
64 bytes from 192.30.255.112: icmp_seq=30 ttl=45 time=324.554 ms
Request timeout for icmp_seq 31
^C
--- github.com ping statistics ---
33 packets transmitted, 28 packets received, 15.2% packet loss
round-trip min/avg/max/stddev = 275.754/315.293/482.425/46.286 ms
```



# 网络抖动和延时

- 光速：30万公里/秒，北京-纽约：1.4万公里，光的一次往返：107ms
- 路由：48 TTL，经过 16 跳路由，每一次都需要处理时间，并且可能会丢包（15% 丢包率）
- 信号转换：光-电信转换
- 其他不可抗拒力：内容审查

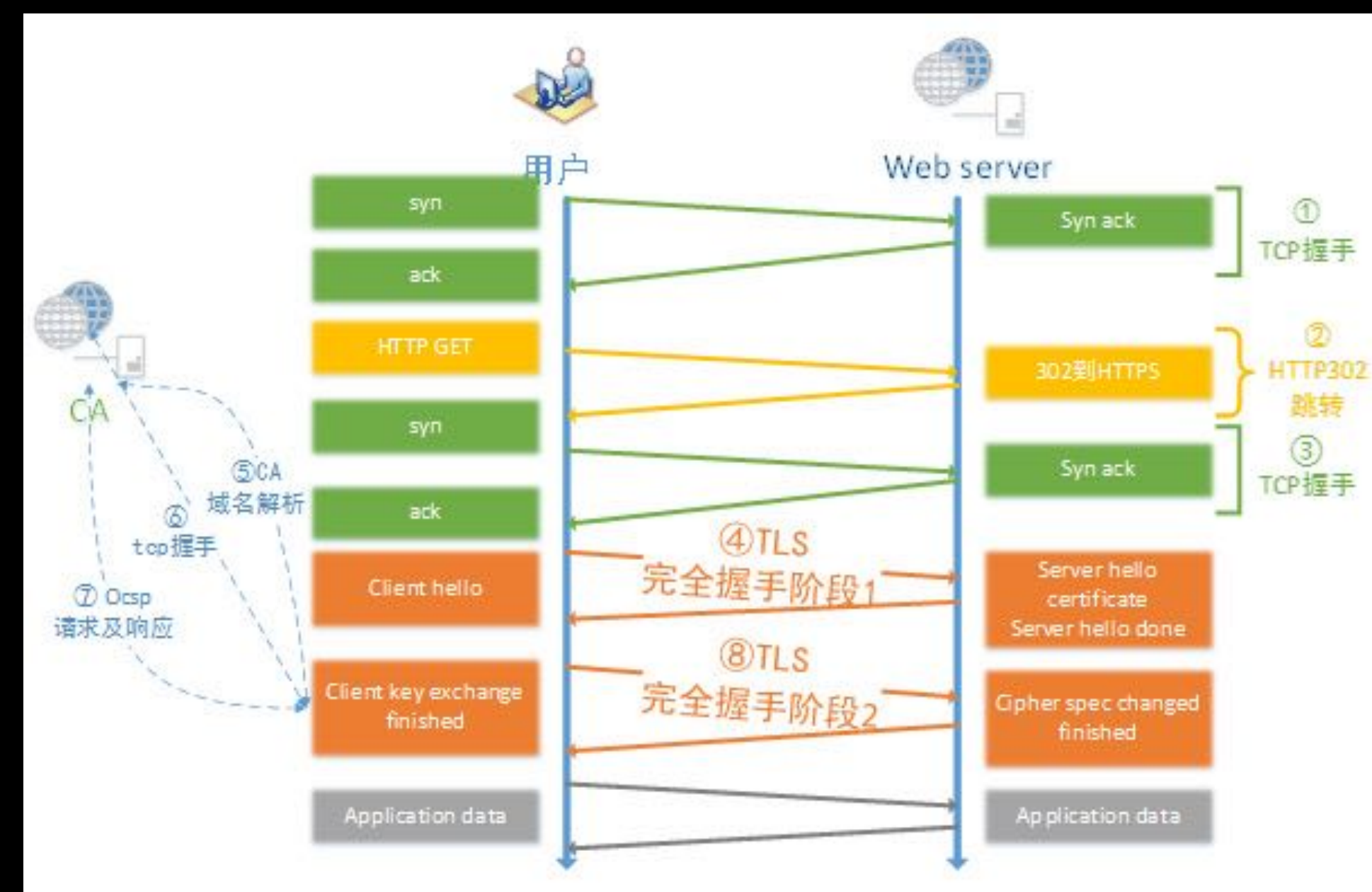
# TCP RTT/RTO

- TCP的超时重传: <https://blog.csdn.net/whgtheone/article/details/80970292>
- $\text{EstimatedRTT} = 0.875 * \text{EstimatedRTT} + 0.125 * \text{SampleRTT}$
- $\text{RTO} = \text{EstimatedRTT} + 4 * \text{DevRTT}$
- 网络抖动会引起丢包重传, 2-5倍 RTT



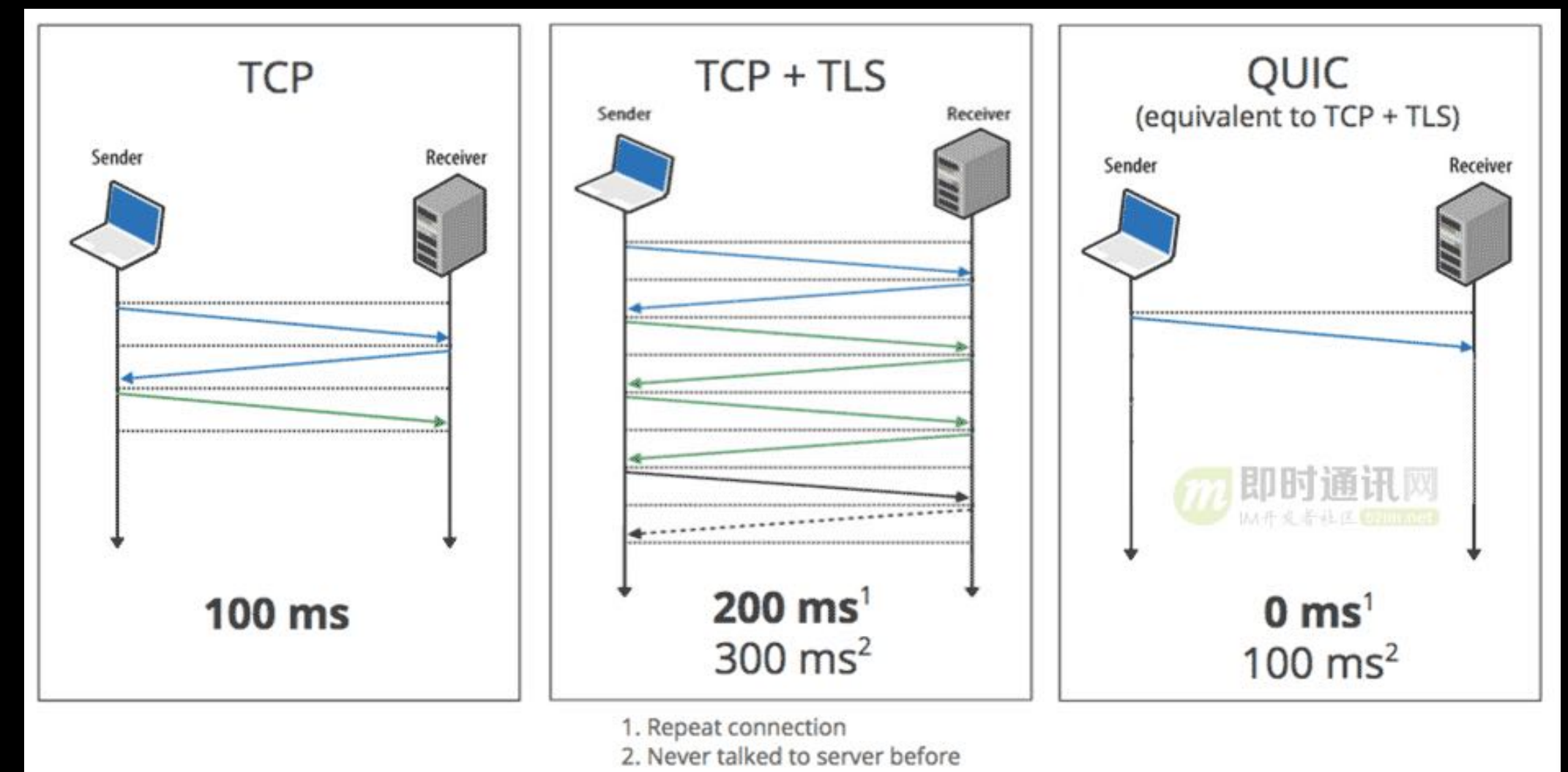
# HTTPS

- TCP 握手 3次通信, 1.5 RTT
- SSL/TLS 握手 4次通信, 2 RTT
- HTTP 请求 & 响应, 1 RTT
- 总计: **4.5 RTT**



# QUIC

- 0-RTT/1-RTT 完成一次加密请求与响应
- IETF 标准
- HTTP3 标准
- Chrome 有 20% 流量使用 QUIC



# HTTP 的通信技术发展史

HTTP1.0	开启 Web 时代
HTTP1.1	KeepAlive
HTTP2	单连接并发
HTTP3	QUIC + HTTP2

# 应用场景

- HTTP3 ， 弱网环境下也可以流畅访问了， 未来3-5年内会普及
- 低延时直播， RTMP over QUIC， 延时从2秒降低到 800ms
- 即时通信（QQ 和微信只是 email ， 不是真正的即时通信）、网络游戏、物联网（设备远程控制和通信）
- QUIC： 高可靠性、低延时的网络通信



# PHP 需要做什么？

Do Nothing



实现 QUIC, HTTP3

HTTP3 + JSON : 单连接并发, 低延时, 抗网络抖动, 流量控制, 主动推送

# 微服务 & 服务治理

# 常见的 RPC 框架

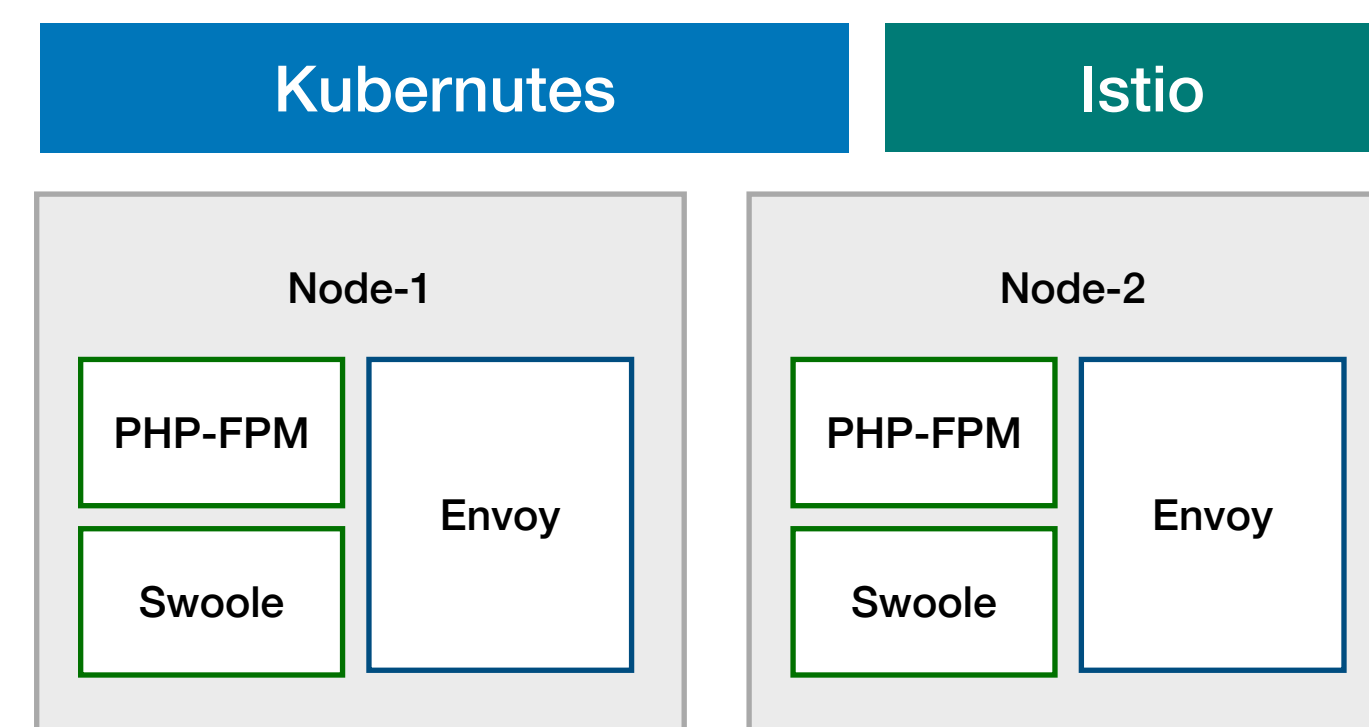
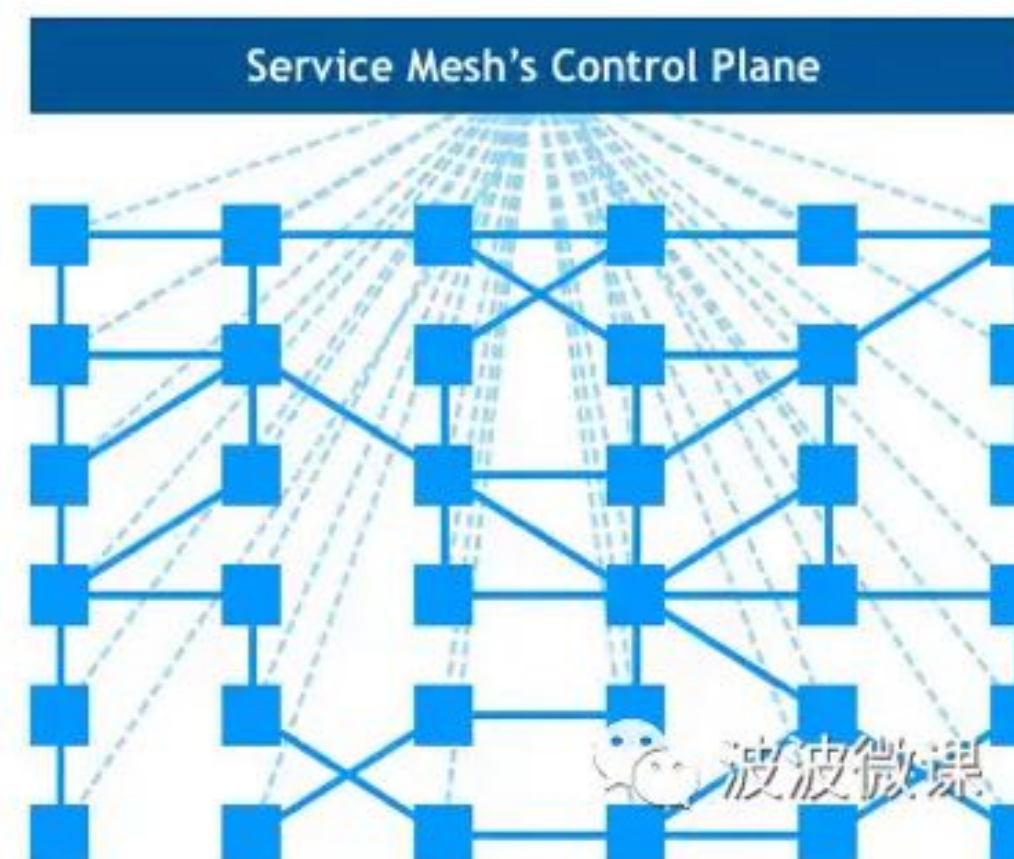
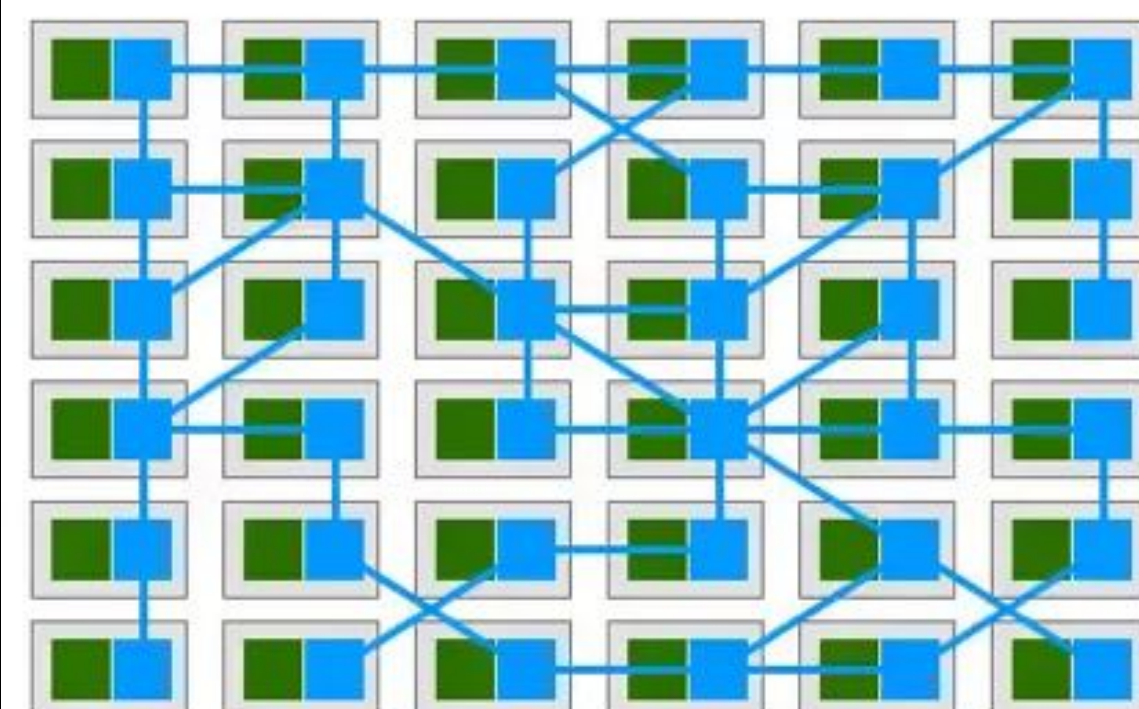
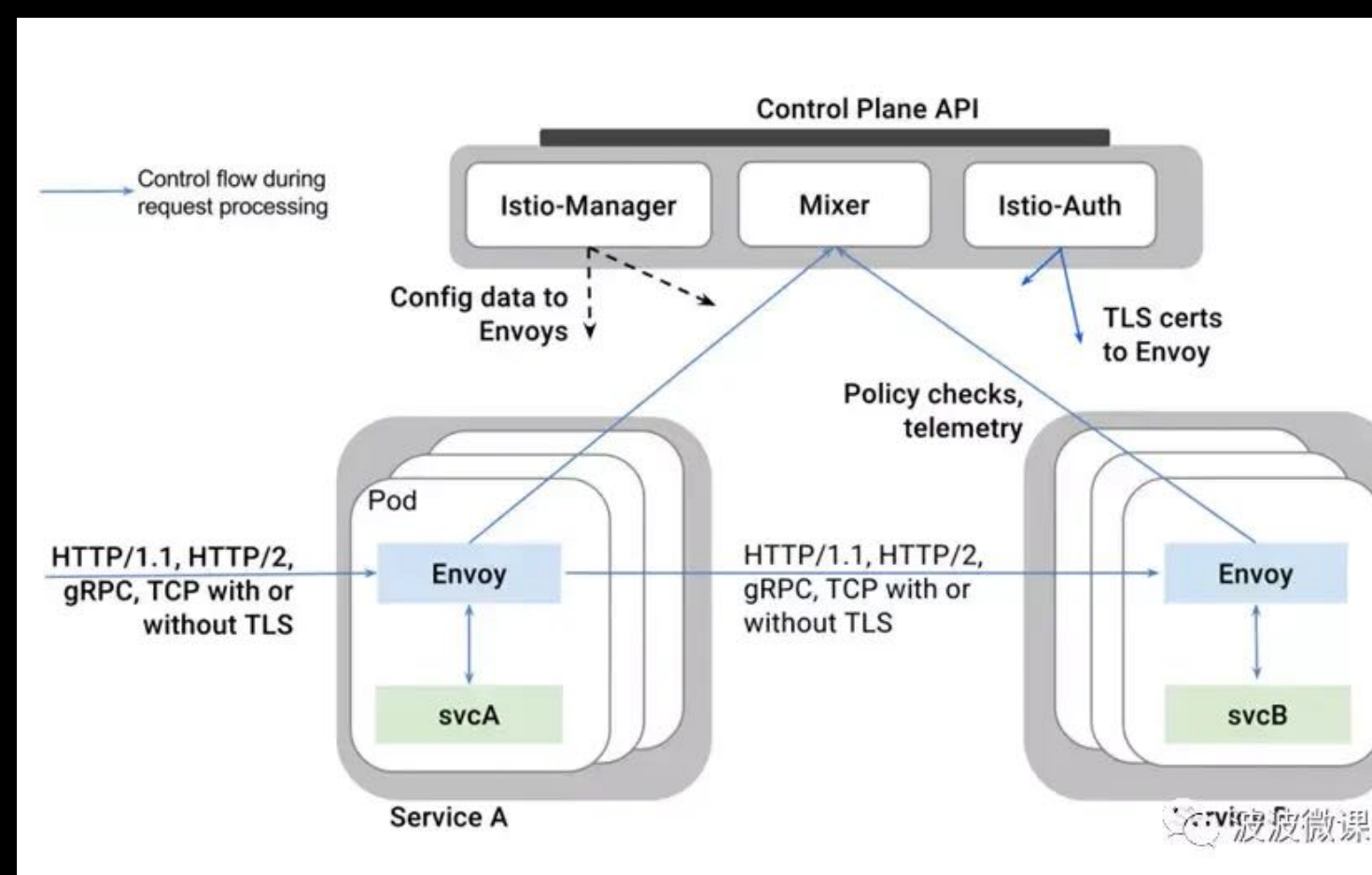
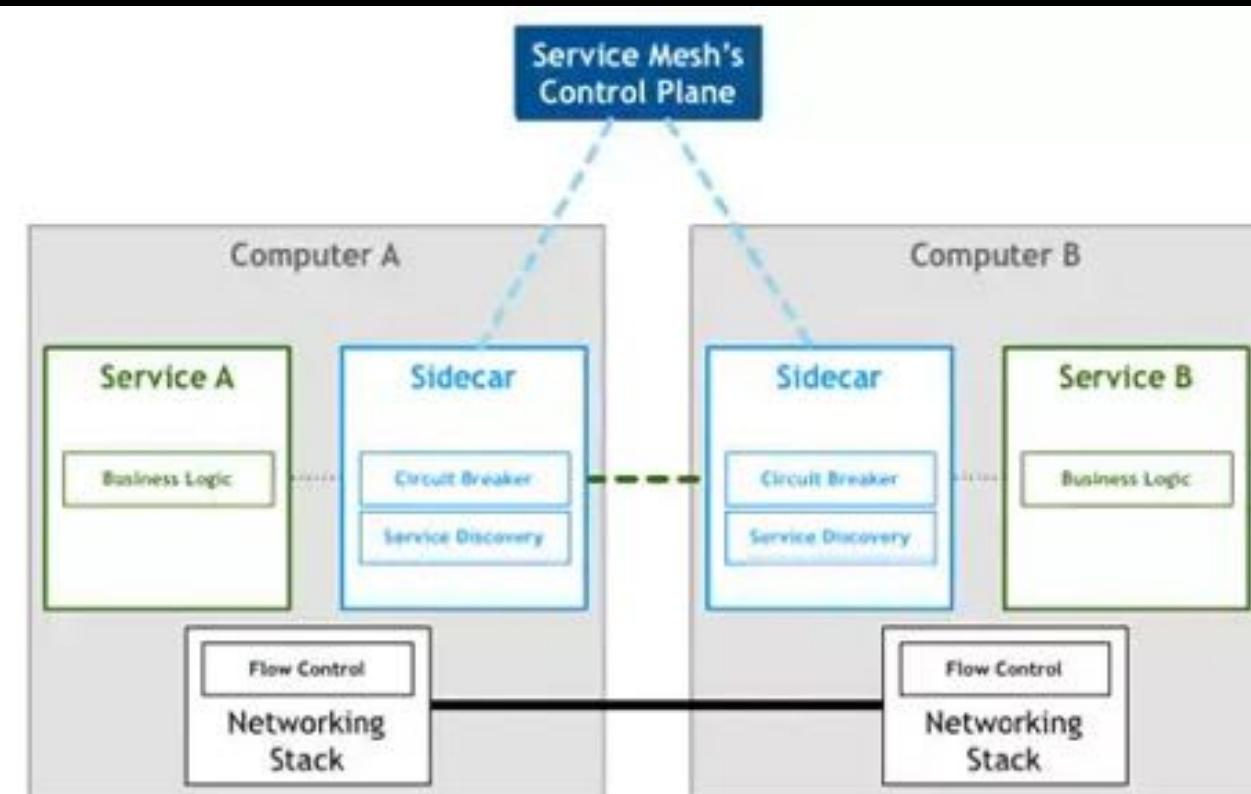
- Spring Cloud: 生态最成熟, 最完善, 只支持 Java
- Dubbo: 成熟稳定, 中文社区支持好, Spring Cloud 之外的另外一个选择, 只支持 Java
- GRPC: 只有 RPC, 缺少整体解决方案, 最接近标准, HTTP2 + ProtoBuf
- Tars: 多语言支持好、配套设施完善、在腾讯有十几年的应用实践
- BRPC: 技术深度最高, 最专业 (知乎: @戈君)
- 其他 ... Swift、Hyperf



# ServiceMesh & ServiceLess

Spring Cloud 向 Service Mesh 转变: <https://www.cnblogs.com/bocloud/p/10895296.html>

# SideCar & Mesh 网格



# Service Mesh 实践

- 控制面板：Istiod
- 数据面板：Envoy
- 容器编排：Kubernetes
- 元数据存储：ETCD
- 流量劫持：iptables、eBPF (Cilium) 、DNS、代理模式

# PHP 如何实现 ServiceMesh ?

Do Nothing



# PHP & ServiceMesh

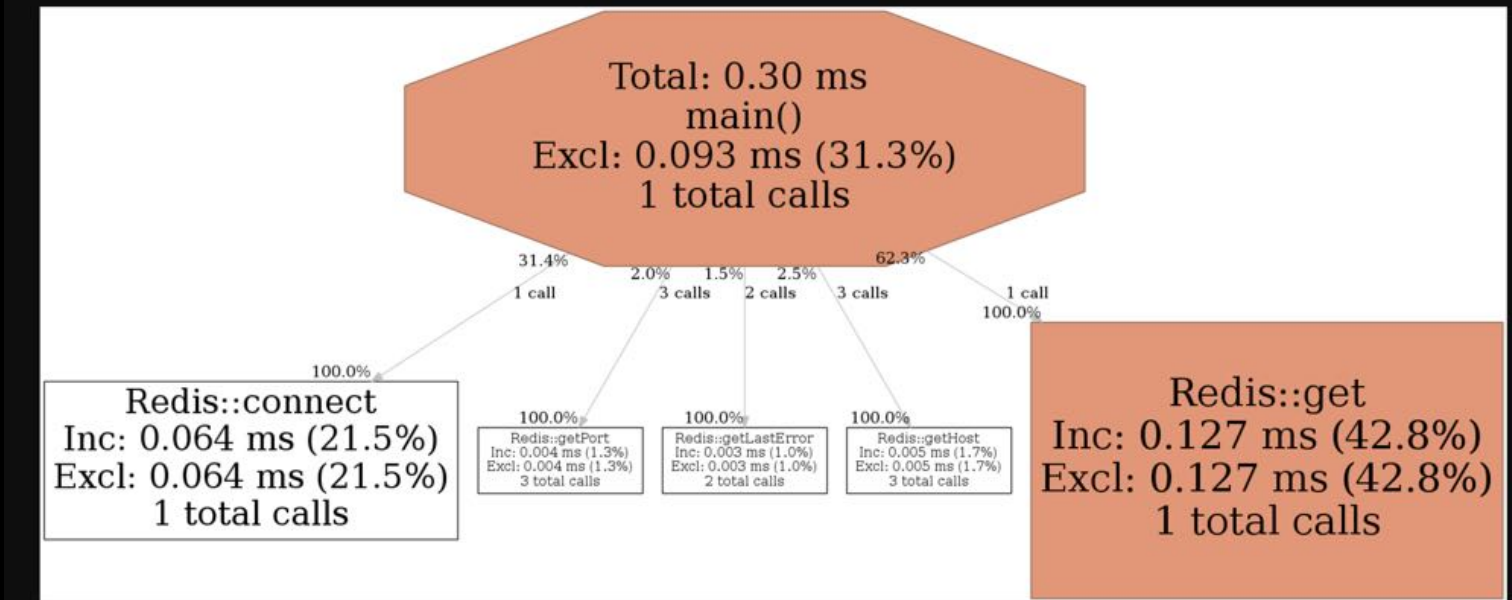
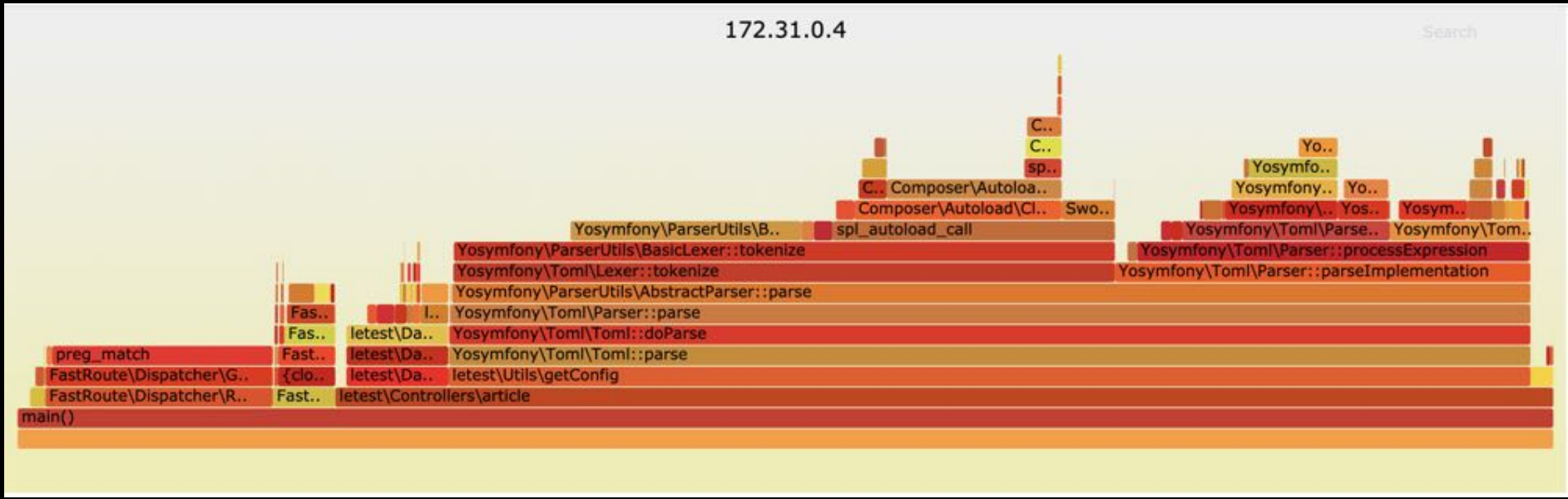
- SideCar 实现了服务发现、故障转移、流量管控、自适应限流、熔断、连接池、协议转换、代理、可观察性 等等所有服务治理的能力
- K8s 管控系统的运行，实现**弹性伸缩**、集群迁移、高可用、混合云架构
- PHP 应用开发者应当专注于深度理解业务、更好地实现业务逻辑，持续提升代码质量，对系统进行抽象和设计，编写单测，编写工具，编写文档，**更优雅、更有效率**
- PHP 内核、扩展、框架、类库，提供更丰富、更易用、更舒适、更高效的编程环境、工具
- 四层架构是20年前的理念，服务架构从二维扁平化变成三维立体化

# PHP 可观察性

- 调用统计：调用次数、QPS、耗时、错误码、主被调关系、拓扑图、历史同比环比、P99 & P95
- 链路追踪：Trace
- 性能分析：性能画像（Profile）、火焰图、阻塞检测、内存泄漏检测
- 推荐一下：**Swoole Tracker**（学而思网校核心业务大规模使用，正在推广到好未来全集团）

项目组 基础服务 -										
接口调用列表										
搜索接口 所有类型 请求目标 2019-12-03 设置										
类型	接口名称	请求目标	调用次数	成功调用	失败调用	平均响应时间(ms)	最小响应时间(ms)	最大响应时间(ms)	失败平均时间(ms)	操作
Redis	Redis::get()	Redis::get	556	556 / 100%	0	0.29	0	4	0	对比   主调   被调
Redis	Swoole\Coroutine\Redis::get()	Swoole\Coroutine\Redis::get	556	556 / 100%	0	0.33	0	7	0	对比   主调   被调
Redis	Redis::connect()	Redis::connect	278	278 / 100%	0	0.51	0	8	0	对比   主调   被调
MySQL	Swoole\Coroutine\MySQL::prepare()	Swoole\Coroutine\MySQL::prepare	278	278 / 100%	0	0.63	0	1	0	对比   主调   被调
MySQL	Swoole\Coroutine\MySQL::Statement::execute()	Swoole\Coroutine\MySQL::Statement::execute	278	278 / 100%	0	0.73	0	3	0	对比   主调   被调
Curl	curl_exec()	curl_exec	278	278 / 100%	0	92.75	11	517	0	对比   主调   被调
MySQL	Swoole\Coroutine\MySQL::connect()	Swoole\Coroutine\MySQL::connect	278	278 / 100%	0	1.27	1	7	0	对比   主调   被调
Redis	Swoole\Coroutine\Redis::connect()	Swoole\Coroutine\Redis::connect	278	278 / 100%	0	0.46	0	2	0	对比   主调   被调
Redis	Swoole\Coroutine\Redis::close()	Swoole\Coroutine\Redis::close	278	278 / 100%	0	0.43	0	1	0	对比   主调   被调

线程详情				
耗时: 143.175 ms, 包含open: 22个, 子调用: 3个				
Service	Type	Name	Duration	时间轴
user_service	Redis	call	143.175ms	
user_service	Curl	php_curl http://demo.swoole-cloud.com/demo/test	131.486ms	
default	Redis	mysql_connect notify_data	0.544ms	
default	Redis	query SET NAMES 'utf8mb4' COLLATE 'utf8mb4_unicode_ci'	0.205ms	
default	Redis	query show tables	0.346ms	
default	Redis	query select 1	0.246ms	
default	Redis	query select 2	0.236ms	
default	Redis	connect connect redis-host.com:6379:0.25	0.122ms	
default	Redis	set set set key.11111111	0.089ms	
default	Redis	get get some key.	0.056ms	
default	Redis	lSize lSize some key.	0.058ms	
default	Curl	php_curl http://10.105.227.58:80/?param=test	0.256ms	
default	Curl	php_curl http://10.105.227.58:80/?param=test3	0.179ms	
default	Curl	php_curl http://10.105.227.58:80/?param=test2	0.179ms	
default	Redis	PDO::prepare show tables	0.005ms	
default	MySQL	PDOStatement::execute show tables	0.43ms	
default	Memcache	addServers addServers array(1)	0.008ms	
default	Memcache	set set info.array(0):0.	0.154ms	
default	Memcache	get get info.	0.058ms	
user_service	Redis	connect connect 127.0.0.1:6379.	0.08ms	
user_service	Redis	get get some key.	0.078ms	
user_service	Redis	get get some key2.	0.054ms	
user_service	Redis	connect notify_data	0.819ms	
user_service	Redis	prepare show tables	0.26ms	
user_service	Redis	execute	0.344ms	
user_service	Redis	connect connect 127.0.0.1:6379.	0.111ms	
user_service	Redis	get get key.	0.089ms	
user_service	Redis	get get key1.	0.064ms	
user_service	Redis	set set key2.11111111.	0.067ms	
user_service	Redis	close close	0.03ms	
user_service	Redis	get 192.168.99.100	0.084ms	
user_service	Curl	php_curl http://10.105.227.58:8822/getAddress?id=192943	0.107ms	
user_service	Redis	PDO::prepare show tables	0.006ms	



**Use somethings, Do not develop somethings**



# PHP8 & Swow

# PHP 生态

- Laravel、Symfony、Composer、Guzzle、Phalcon、Zend-Expressive
- WordPress、MediaWiki、Drupal
- PHPStorm、Xdebug、PHPUnit
- Swoole、Hyperf、EasySwoole、MixPHP、Swift、IMI、Saber ...
- Swoole Tracker、Swoole Compiler

# Swoole

- 趋于稳定成熟，持续提升性能和稳定性
- 代码质量、代码规范、文档、测试

## Mastering Swoole PHP

Build high performance concurrent system with async and coroutines

PRE ORDER FOR £24.96 £48

## Mastering Swoole PHP

Build concurrent system with async and coroutines

Pre Order

## Mastering Swoole PHP

Build concurrent system with async and coroutines

- General Concepts
- Linux OS
- Asynchronous I/O
- Processes
- Reactor pattern and internal
- Coroutines and PHP
- The names and code examples

# Swow

- 基于 libuv
- 支持 Windows 和 PHP-FPM
- 语言级别协程
- 纯 C 实现，未来考虑推动合并到 php-src
- 微内核，最小核心，library 完全使用 PHP 代码编写

# PHP

- 近 20 年内无数的商业技术死亡（Delphi、VB、Flash），这些商业技术一旦无法为企业贡献价值，就会被抛弃
- PHP 没有商业公司控制，它是真正民主、开放的开源项目和社区，人人可以参与，它是**生生不息的、薪火相传的、持续进化的**
- PHP8、Swow
- 新生代：Nikic、Twosee

# 个人感悟



- Stay hungry, Stay foolish
- 空杯心态，持续学习和探索
- 对线上系统保持敬畏

# Thanks

