Skynet

基于 Actor 模式的开源框架

众核时代的并行编程

- “The Free Lunch Is Over”

  – Herb Sutter , 2005
- E5420 (2.5GHz) 2004 vs E5-2620v3 (2.4GHz) 2014
- 我们获得了更多的核心、更多的线程、更大的缓存、更大的带宽
- 我们得不到更高的主频。

Erlang

- 1998 年 Erlang 开源
- 2006 年传入国内 C++ 程序员圈
  - Erlang China User Group 始于 2007
  - 现改名为 Effective Cloud User Group
- 并行和分布式
  - Erlang 采用的 actor 模式使其擅长并行处理
  - 错误处理机制和储存管理为分布式服务
  - Erlang 并不擅长储存密集型数值计算

Actor 模式

- 一切都是 Actor
  - 一切都是 Object？
  - 并行的面向对象模式！
  - 创建 actor / 处理消息 / 发送消息
  - 发送的消息和发送者解耦、异步通讯。
- 没有 actor 的语言提供的框架：
  - Akka by Scala
  - CAF: C++ Actor Framework by C++
  - Remact.Net by .net

- https://github.com/cloudwu/skynet
- 中文文档： https://github.com/cloudwu/skynet/wiki
- 邮件列表： skynet-users@googlegroups.com
- QQ 群： 340504014

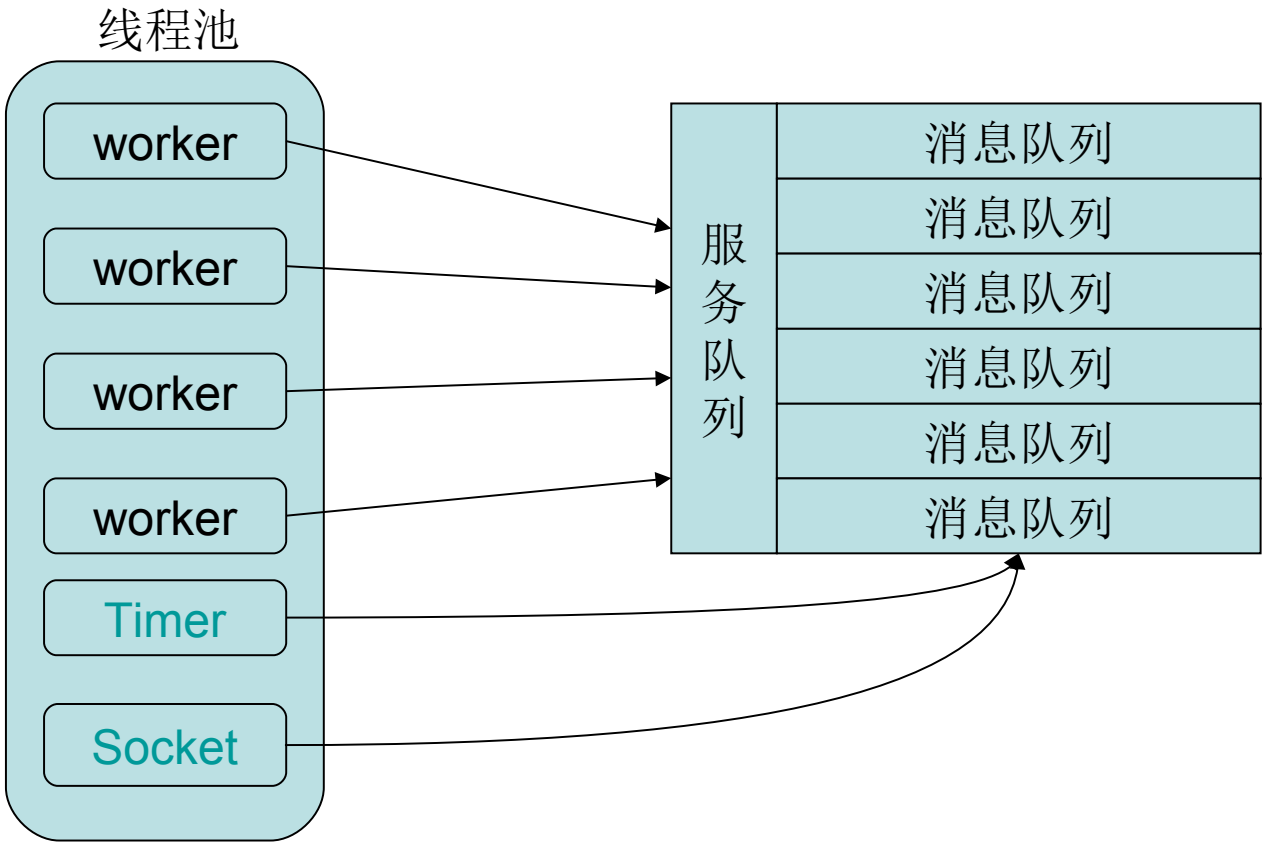
历史

- 2010 年 想法
- 2011 年 12 月 基于 Erlang 的第一版实现
- 2012 年 7 月 基于 C 重写
- 2012 年 8 月 1 日 开源发布
- 2014 年 4 月 22 日 v0.1.0
- 2014 年 11 月 28 日 v0.9.1
  - 896 commits
  - 21 contributors
  - 1925 stars
  - 955 forks
  - 1387 qq 群用户 , 316 mailling list 订阅者

Code Base

- ~ 5000 行 C 核心代码
  - 消息分发
  - Actor 调度
  - Timer 管理
  - 基于 epoll/kqueue 的 socket 库 (支持 tcp/udp)
- ~ 1000 行 C 核心服务代码
- ~ 1000 行 Lua 核心库
- ~ 5000 行 Lua 外围库
  - Redis / Mysql / MongoDB driver
  - crypt , sproto , sharedata , etc.
- 单进程 + 可选 Lua 沙盒
- 可选分布式结构
- MIT License

消息调度模块



Xeon(R) CPU E5310 @ 1.60GHz
echo 服务处理能力每秒超过 0.5M 条（短）消息

异步编程

- Coroutine vs callback
- lua 5.2 coroutine 的内存开销仅 208 字节
  - C 线程很难减少栈的内存开销
- coroutine 对异常有良好的支持
  - javascript 需要 Promise 模式
- 复用 coroutine 避免过频的 GC
- 远程过程调用 (RPC)
  - 小心异步过程中的状态改变
- 快速失败模式

Actor 沙盒

- C 模块 (动态库 so)
- Lua State != Erlang Process
- Lua 沙盒隔离（ 30 + 20 K 内存占用）
- Lua 5.2 (Lua JIT 2.0 可选)
  - 命令式语言
  - 轻量 coroutine
  - 良好的 C 交互性能
- Lua 库 + 服务
  - 异步 socket 库
  - Launcher
  - DB Driver
- 共享数据 (sharedata / STM)

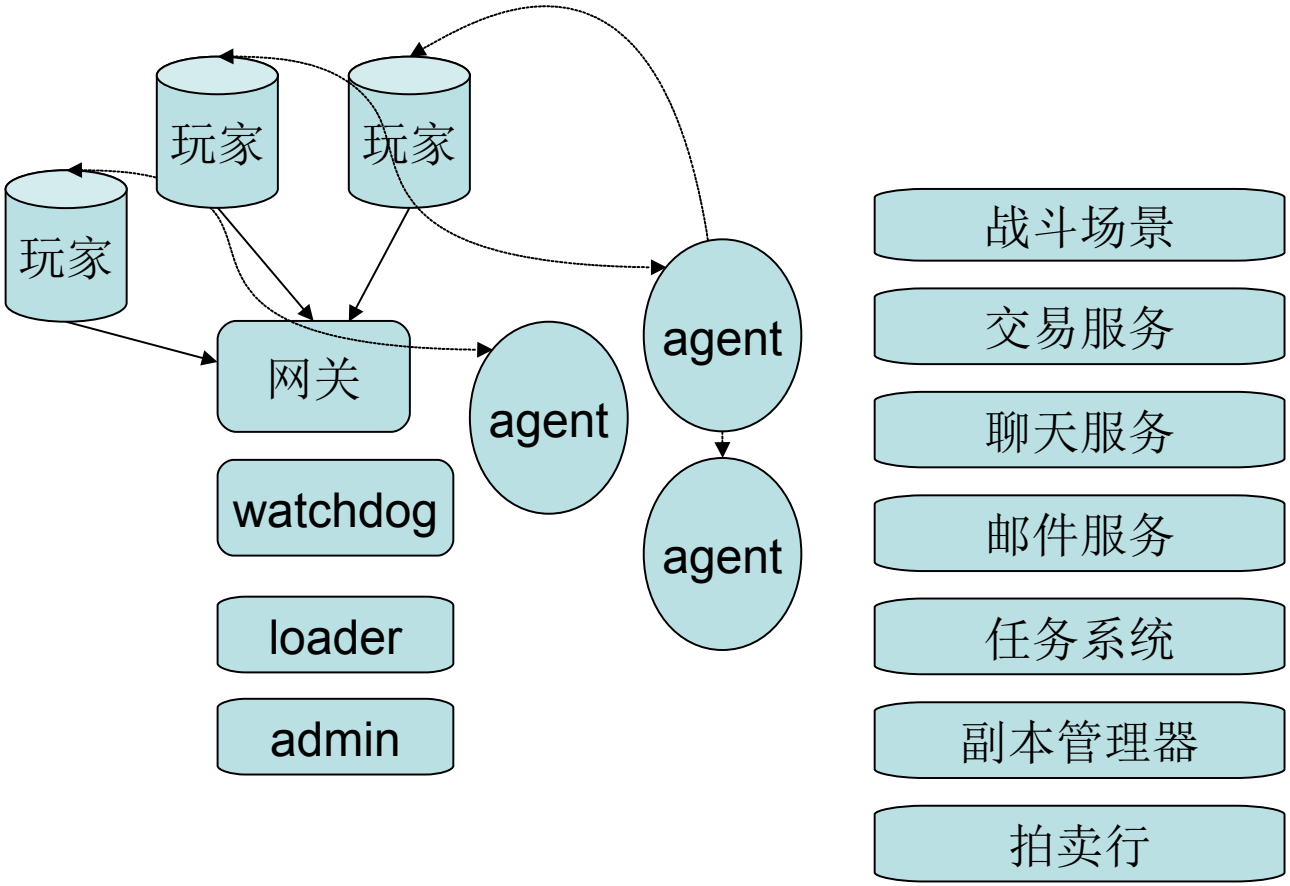
通讯协议

- 进程内消息传递
  - 文本协议 （C 服务 ）
  - 自定义序列化库 （Lua 服务 ）
  - 内存数据结构 （ 自定义 ）
- 跨进程消息传递
  - 自定义协议
  - sproto
  - google proto buffers https://github.com/cloudwu/pbc
  - json , etc.
- 广播和组播

分布式解决方案

- skynet 支持两种分布式方案
  - harbor 模式用于拓展计算能力的不足
  - cluster 模式提供弹性
  - 可以一起使用
- 如非必要、在一台机器解决
  - 使用 cluster 做弱关联
- 不做热更新、只做热修复
  - A/B 滚服，定期维护，减少复杂度。

MMORPG


玩家
玩家
玩家
网关
agent
agent
agent
watchdog
loader
admin
战斗场景
交易服务
聊天服务
邮件服务
任务系统
副本管理器
拍卖行

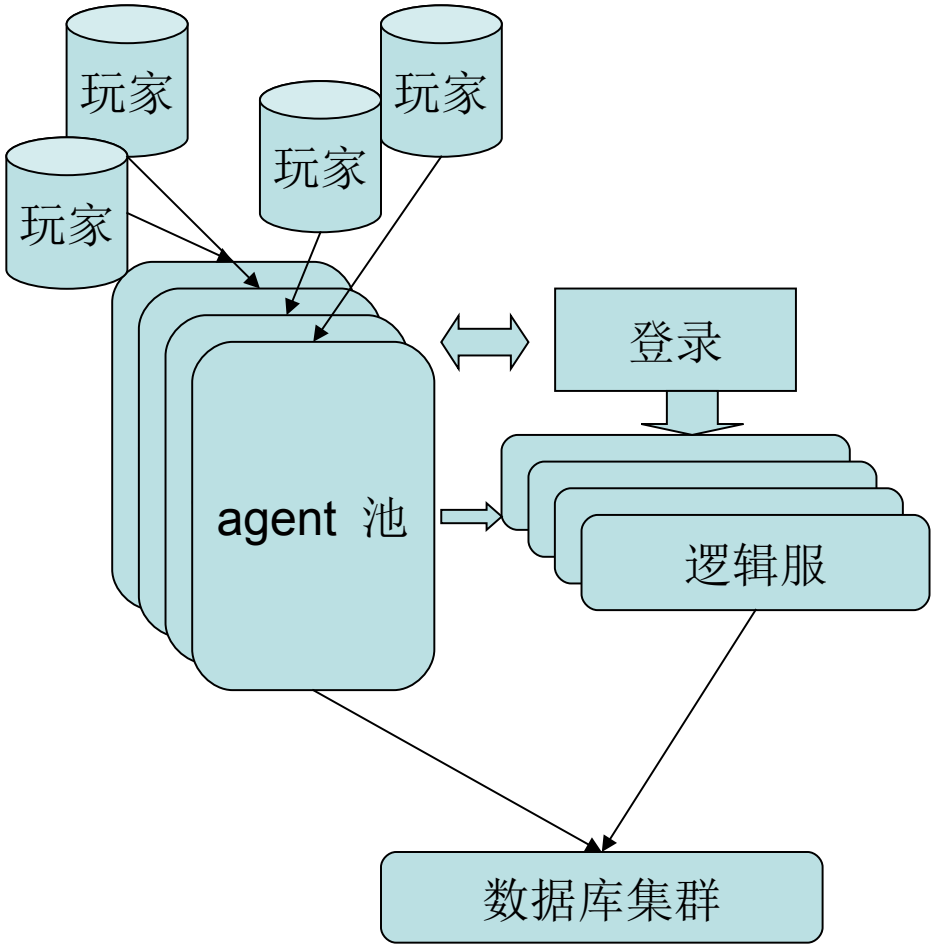
MMORPG 服务器性能

- E5-2620v1 (2.0 GHz) * 2 , 32G RAM
  - 开启超线程，系统可见 24 个核心
- 10k 个 tcp 长连接
- 200 人相互面积伤害，延迟在可接受范围
  - 计算复杂度 O(n^2) Lua 编写
  - CPU 平均开销 0.5% / 人 ，系统上限 2400 %
  - < 5M Ram / 人
- 多核的优势在于处理时间平滑

陌陌争霸

- E5-2420v1 到 E5-2650v3
- 初期压力测试单机 20K 用户
  - 承载能力随业务复杂而下降
  - 承载能力随核心数增加线性上升
- 全服峰值接近 200K 用户
  - 登录操作无明显延迟
  - 对手搜索（全局单点）无明显延迟

典型手游集群



- 不按硬件分服
- 玩家在登录处获取令牌
- 玩家登录任意 agent 池
- 逻辑服处理排行榜
- 根据运营需要在线调整
- 统一使用数据库集群
- 避免单点

调试和优化

- 内建性能分析模块
- Lua 模块内建监控协议
- 替换 CRT 内存管理库 (jemalloc)
- 进程内消息传递减少拷贝
- 优化向自身发送的消息
- 合并 timer 请求
- 高性能要求的服务使用 C/C++ 编写（慎用）
- 为 Lua 编写 C 模块
  - AOI
  - 寻路
  - 组播
  - 公式计算
- 优化登录、找到热点、避免单点。

谢谢

http://blog.codingnow.com

新浪微博：@ 简悦云风