

**KEDACOM**

# OSP——从掌握到精通

吴彬  
2011-04-28

苏州科达科技有限公司

- 通过本课程的学习，您将能够：
  - 了解 OSP 的作用
  - 了解OSP 的框架和实现原理
  - 掌握 OSP 使用方法和要点
  - 明白 OSP 的不足与缺陷
  - 了解OSP的发展方向

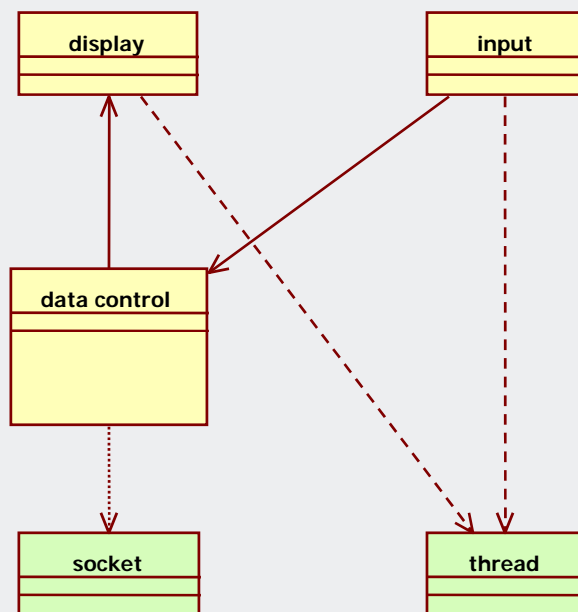
- 本课程适用于：  
    OSP初学者  
    对OSP使用、实现有兴趣的开发人员

- OSP 的功能
- OSP 的架构
- OSP 的实现原理
- OSP 的使用要点
- OSP 的改进方向

**OSP为我们带来了什么？**

- 编写一个聊天程序
- 使用C++
- Windows Console
- 基于TCP





- Input 负责接收用户输入
- Display负责呈现
- Data control负责数据和链接的管理



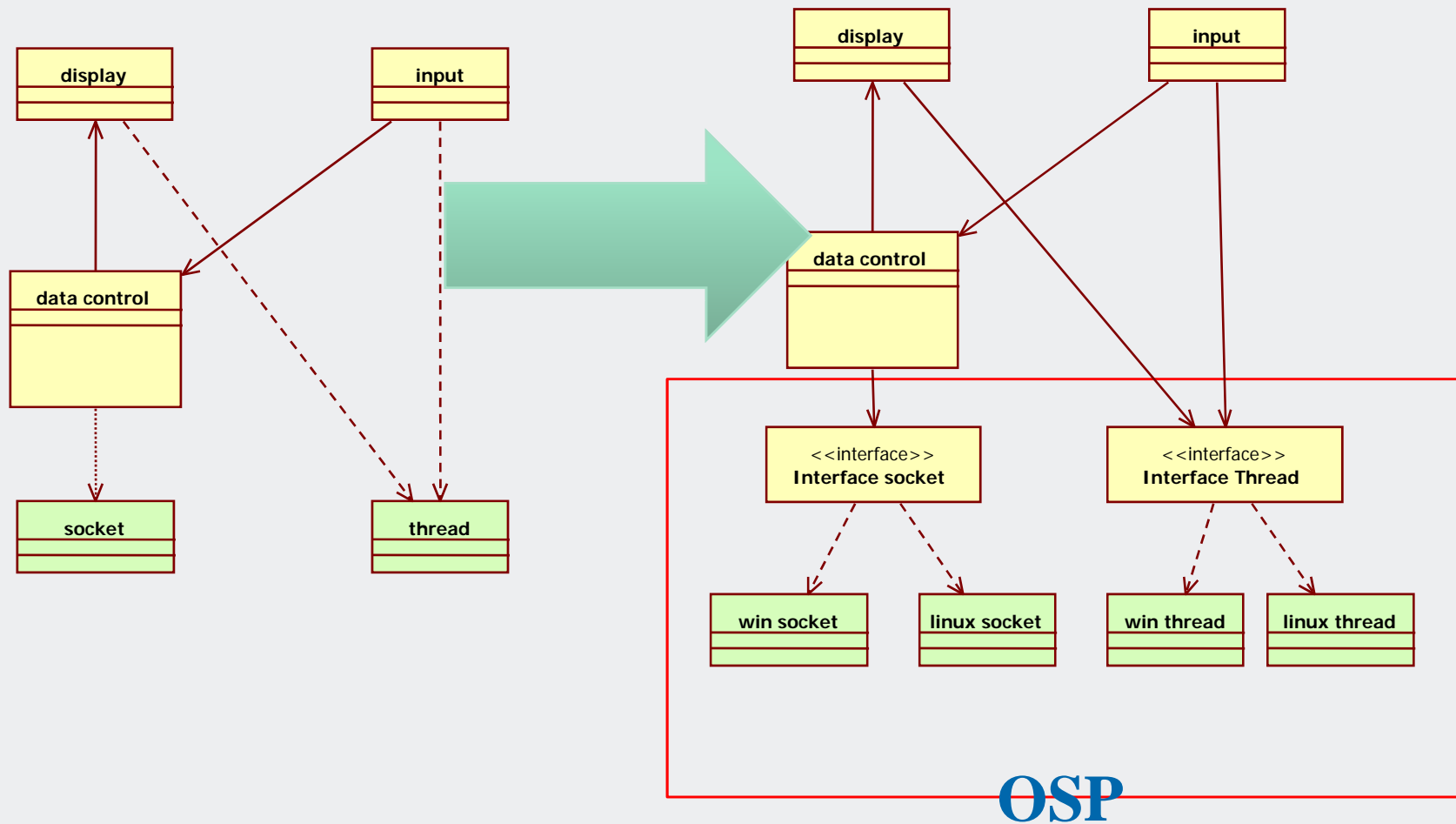
- 多线程管理
- 资源锁
- connect accept send recv....
- 异常断链处理

- 编写一个聊天程序
- 使用C++
- **Linux Console**
- 基于TCP

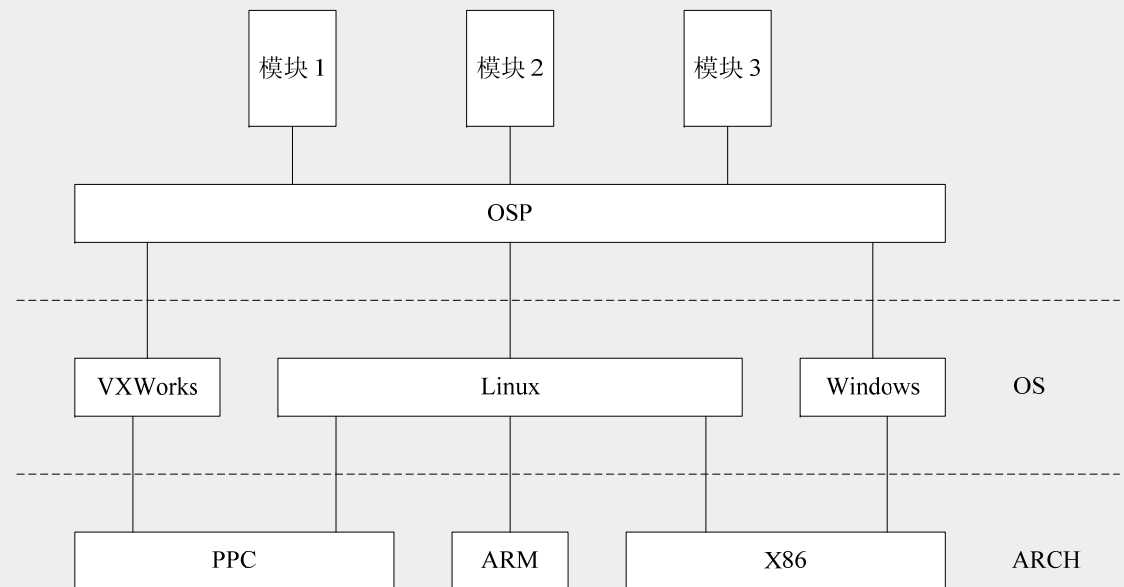


# 入职新任务2

KEDACOM



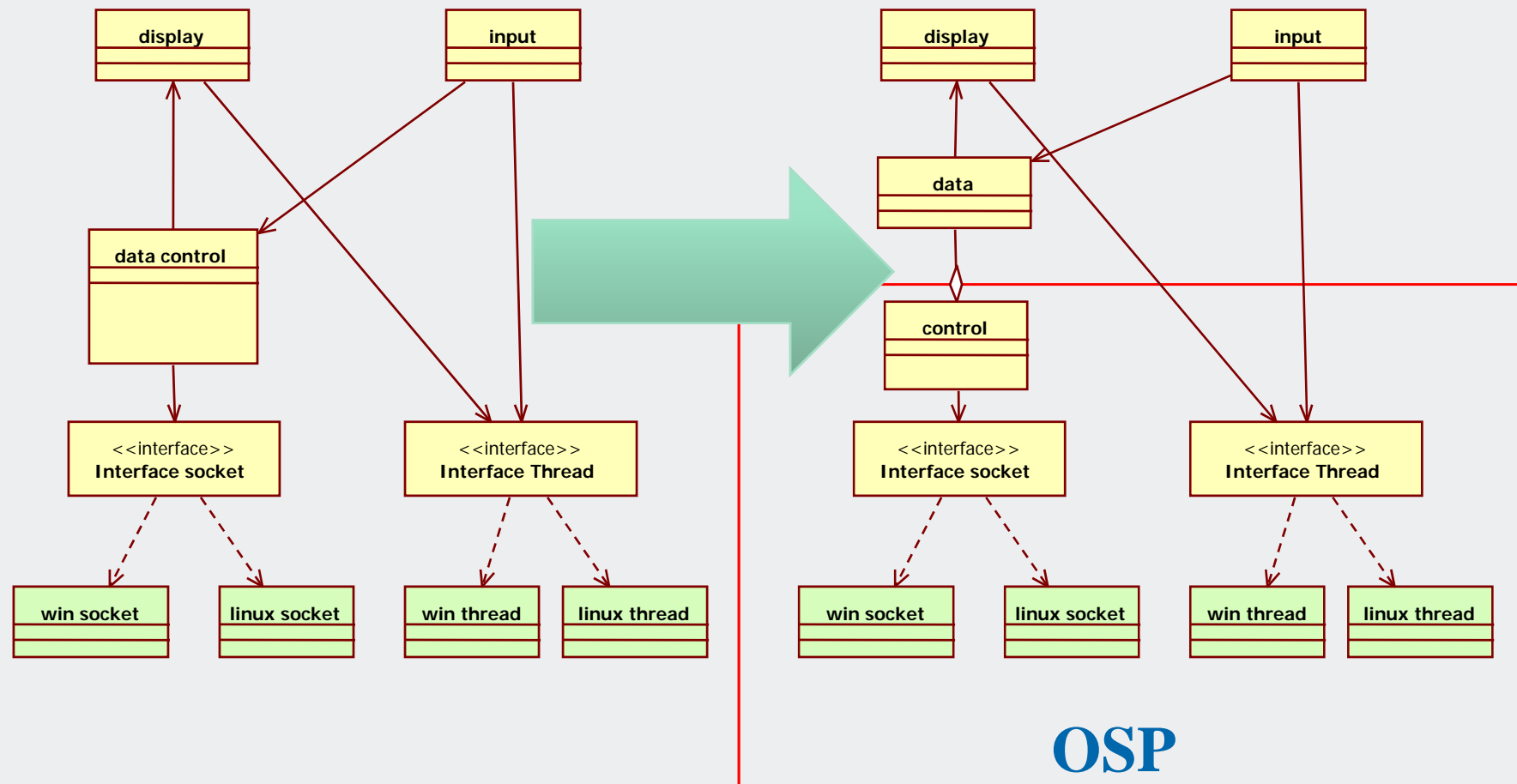
- OspTaskCreate
- OspSemCCreate
- OspPrintf
- OspDelay
- OspTickGet
- OspAllocMem



**OSP还有什么？**

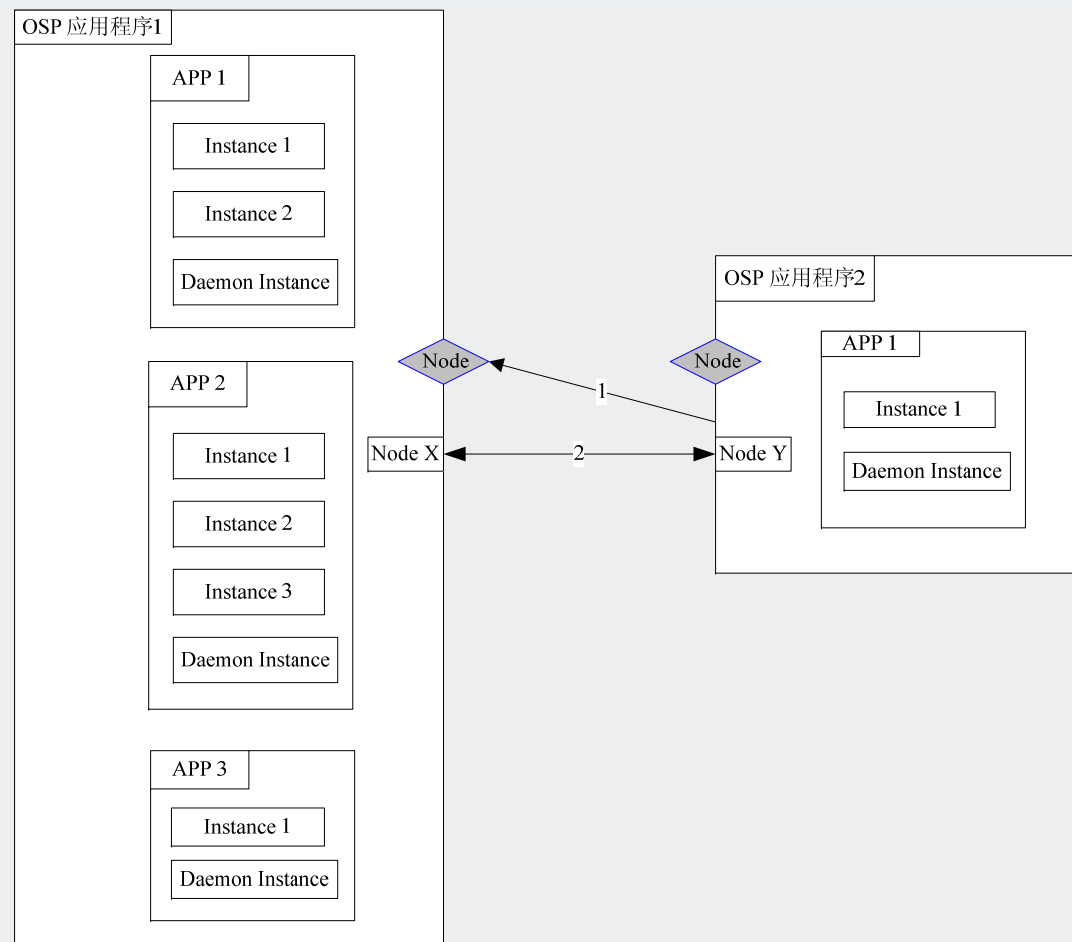
# 不仅仅是封装

KEDACOM



# OSP TCP消息框架

KEDACOM



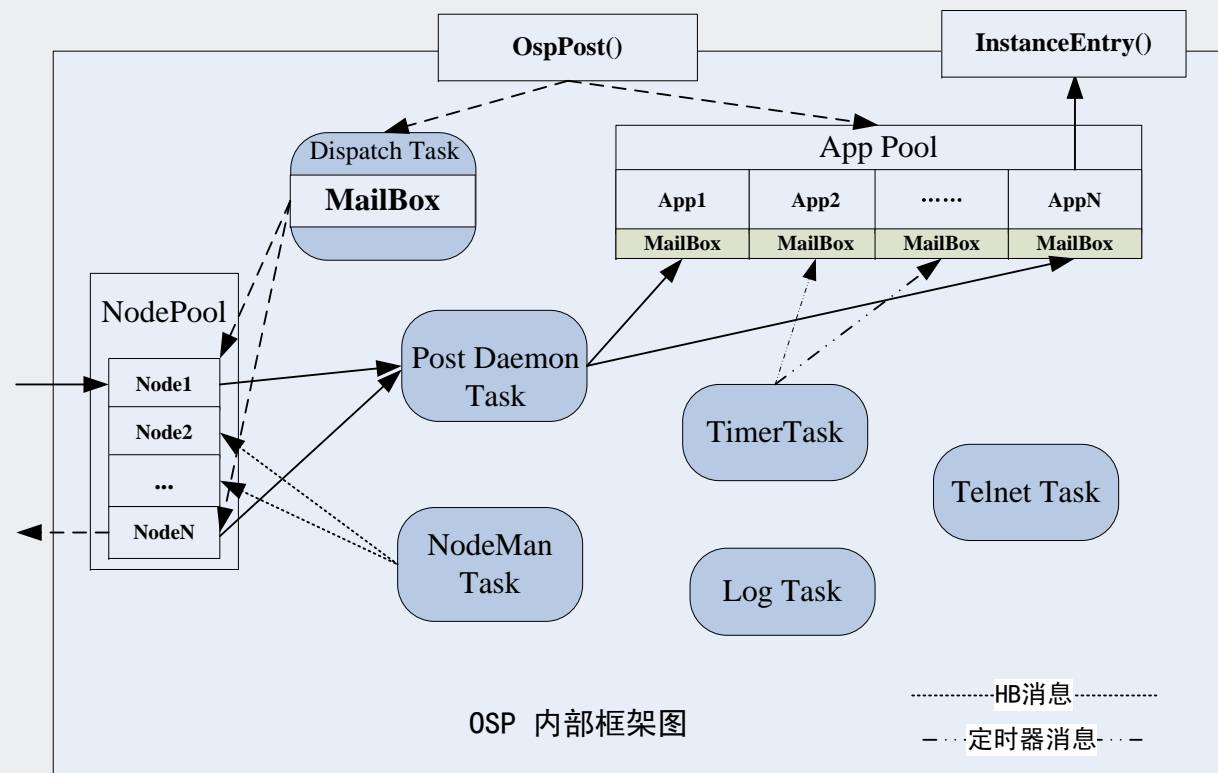
- 定时器
- 断链侦测
- 日志
- telnet远程调试



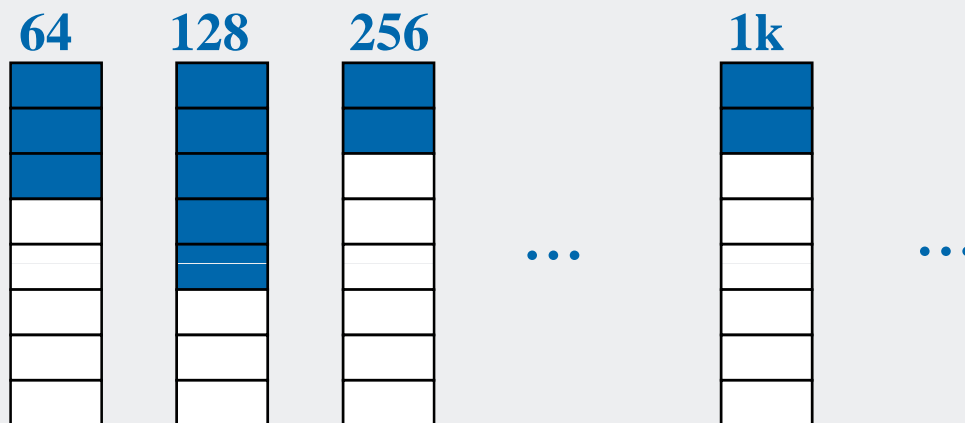
# OSP的实现

# OSP内部框架

KEDACOM

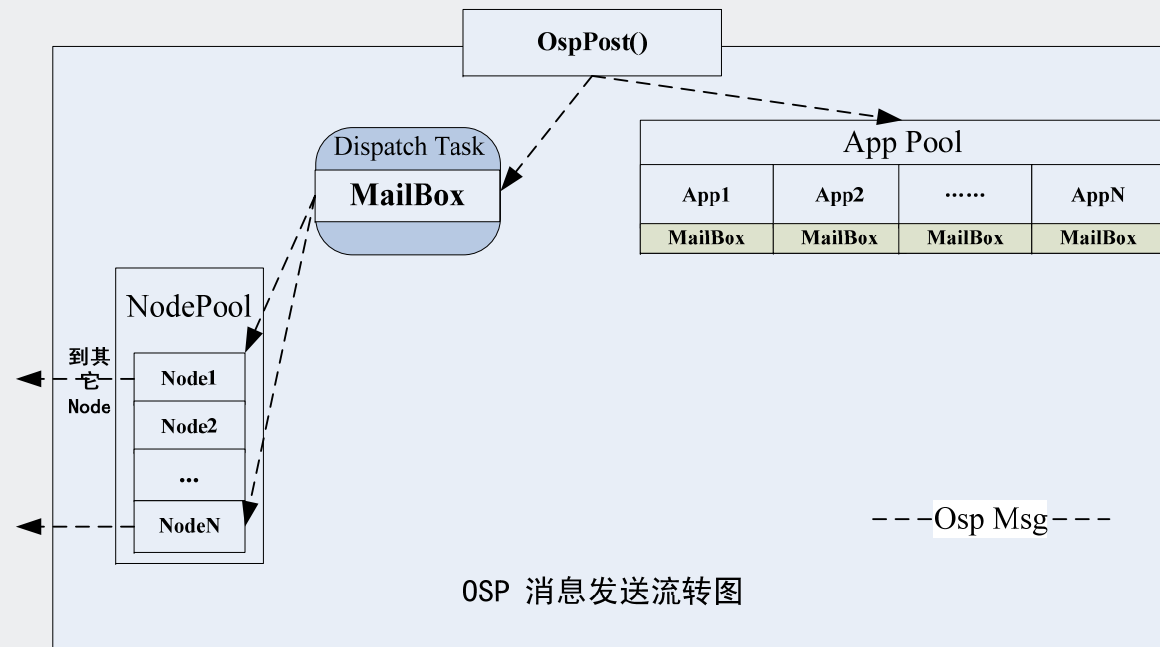


- 目的：避免重复 malloc和free。
- 内存池：OSP维护一组内存块队列，每个队列维护一些固定大小的内存块，内存块大小从64B到4MB成倍递增。
- [OspAllocMem] 按需要分配的大小选择合适的队列进行分配。如果队列中有空闲块，则设置块标记；否则malloc新块，设置块标记，并加入队列。
- [OspFreeMem] 不会free，仅复位块标记。
- [OspQuit] 不管内存块是否被回收（设置块标记），都会被free。



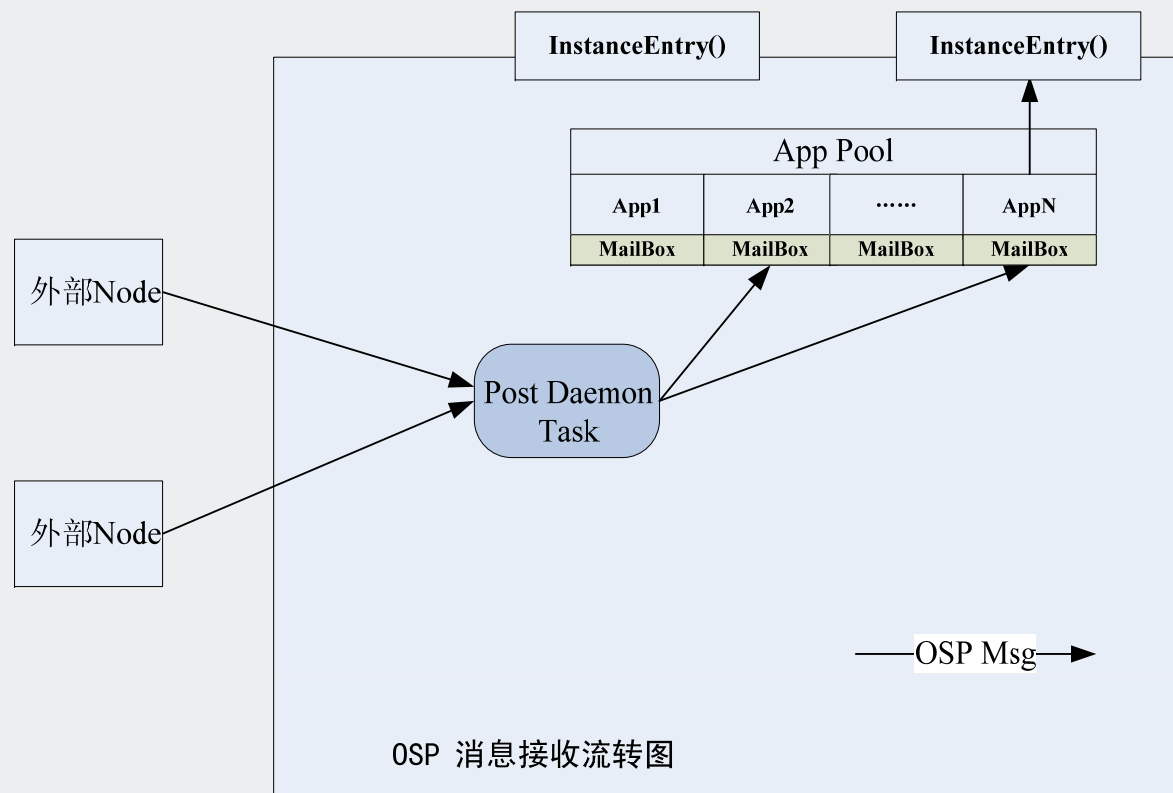
# OSP消息发送

KEDACOM



# OSP消息接收

KEDACOM



# 如何使用OSP

- 代码
- 调试

## OSP的改进



- OSP的瓶颈在何处？
- OSP有什么潜在的问题？

# KEDACOM



- Node邮箱
- 消息优先级
- 消息压缩

- send与备用邮箱机制  
一旦在send阻塞时，有超过备用容量的消息到达时，该APP线程将会永久性阻塞！

**提问时间**

谢谢！